

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**

THIS PAGE BLANK (USPTO)

(19)

BUNDESREPUBLIK

DEUTSCHLAND



DEUTSCHES

PATENTAMT

(12)

Übersetzung der europäischen Patentschrift

(87)

EP 0 246 666 B1

(10)

DE 37 89 728 T 2

(51)

Int. Cl.⁵:

G 06 F 13/36

H 04 L 12/40

G 06 F 13/38

G 06 F 13/42

F 02 D 41/26

G 06 F 13/374

(21)

Deutsches Aktenzeichen:

37 89 728.4

(86)

Europäisches Aktenzeichen:

87 107 565.1

(86)

Europäischer Anmeldetag:

22. 5. 87

(87)

Erstveröffentlichung durch das EPA:

25. 11. 87

(87)

Veröffentlichungstag

der Patenterteilung beim EPA:

4. 5. 94

(47)

Veröffentlichungstag im Patentblatt:

15. 12. 94

DE 37 89 728 T 2

(30)

Unionspriorität:

(32)

(33)

(31)

22.05.86 US 866628

22.05.86 US 866629

22.05.86 US 866630

(73)

Patentinhaber:

Chrysler Corp., Highland Park, Mich., US

(74)

Vertreter:

Hauck, H., Dipl.-Ing. Dipl.-Wirtsch.-Ing., 80336 München; Graalfs, E., Dipl.-Ing., 20354 Hamburg; Wehnert, W., Dipl.-Ing., 80336 München; Döring, W., Dipl.-Wirtsch.-Ing. Dr.-Ing., 40474 Düsseldorf; Siemons, N., Dipl.-Ing. Dr.-Ing., Pat.-Anwälte, 20354 Hamburg

(84)

Benannte Vertragsstaaten:

DE, FR, GB, IT, SE

(72)

Erfinder:

Miesterfeld, Frederick O.R., Troy Michigan 48084, US; Fassnacht, Ronald E., Rochester Michigan 48063, US; McCambridge, John M., Northville, MI 48167, US; Nasiadka, Jerry M., Warren Michigan 48093, US

(54)

Serieller Datenbus für verschiedene Betriebsarten (SCI, SPI und gepufferte SPI) und Verfahren für serielle Peripherieschnittstelle in einem seriellen Datenbus.

Anmerkung: Innerhalb von neun Monaten nach der Bekanntmachung des Hinweises auf die Erteilung des europäischen Patents kann jedermann beim Europäischen Patentamt gegen das erteilte europäische Patent Einspruch einlegen. Der Einspruch ist schriftlich einzureichen und zu begründen. Er gilt erst als eingelegt, wenn die Einspruchsgebühr entrichtet worden ist (Art. 99 (1) Europäisches Patentübereinkommen).

Die Übersetzung ist gemäß Artikel II § 3 Abs. 1 IntPatÜG 1991 vom Patentinhaber eingereicht worden. Sie wurde vom Deutschen Patentamt inhaltlich nicht geprüft.

DE 37 89 728 T 2

EP 87 107 565.1-2201
CHRYSLER CORPORATION

M-6974-N

BESCHREIBUNG:

Die Zielsetzung für den offenbarten seriellen Datenbus liegt darin, daß mehrere Mikroprozessoren leicht miteinander über ein gemeinsames Paar Drähte kommunizieren können, wobei ein einer Telefonteilnehmerleitung ähnliches Schema benutzt wird. Die Erfindung betrifft die Datenkommunikation zwischen Teilen eines multiplexen Systems in einem Fahrzeug. Der serielle Datenbus und die integrierten Schnittstellen, die hier erläutert werden, sind in der Lage, viele Arten von Kommunikationsprotokollen zu tragen.

Der Zweck des offenbarten Verfahrens für eine serielle periphere Schnittstelle (SPI) in einem seriellen Datenbus liegt darin, ein Verfahren zu schaffen, mit dem ein SPI-Kanal als Teil einer seriellen Datenbus-Schnittstelle IC bedienbar ist.

Der Zweck des offenbarten Verfahrens für eine gepufferte serielle periphere Schnittstelle in einem seriellen Datenbus ist es, ein Verfahren zu schaffen, mit dem ein gepufferter SPI-Kanal als Teil der hier beschriebenen seriellen Datenbus-Schnittstelle IC bedienbar ist.

Datenkommunikationen zwischen Mikroprozessoren oder Mikrocomputern sind im Stand der Technik vielfach bekannt.

Solche Mikroprozessoren oder Mikrocomputer lassen sich über lokale Netzwerke (LAN) verbinden, wobei einer der Mikrocomputer die Steuerung eines seriellen Datenkanals übernimmt, der gemeinsam an alle anderen Mikroprozessoren des LAN angeschlossen ist, um Daten zu irgendeiner anderen Einheit zu

übertragen. Die hierfür erforderlichen Protokolle, Steuergeräte und Software sind sehr aufwendig, insbesondere bei großen Systemen.

Bei einem Fahrzeug handelt es sich um ein kleines Anwendungsgebiet, so daß die komplexen Leistungsfähigkeiten nicht nötig sind, die ein LAN bietet.

Mit bekannten digitalen Datenbussen hat man die vorbeschriebene Datenkommunikation in kleineren Anwendungsgebieten durchgeführt. Ein solches System ist erläutert in SAE Paper No. 840317 von Ronald L. Mitchell "A Small Area Network for Cars". Auf dieses Dokument wird hier ausdrücklich Bezug genommen. Ein weiterer digitaler Datenbus ist in U.S. Patent 4,429,384 von Kaplinsky erläutert "Communication System Having an Information Bus and Circuits Therefor".

Ferner werden Entwicklungen auf diesem Gebiet im SAE Paper No. 860390 von Frederick H. Phail und David J. Arnett "In Vehicle Networking - Serial Communications Requirements and Directions" beschrieben.

Die vorliegende Erfindung unterscheidet sich von diesem Stand der Technik dadurch, daß sie eine konstante Geschwindigkeit benutzt, kein Quittungsbit braucht und keine Anforderungen an eine feste Verknüpfung zwischen der Sendestation und der Empfangsstation benötigt. Ebenfalls ist die Verknüpfung zwischen dem Nachrichtensender und dem Empfänger erfindungswesentlich.

Zu Kollisionsdetektoren in Datenkommunikationssystem wird auf folgende U.S. Patente hingewiesen: US 4,281,380; US 4,409,592; US 4,434,421; US 4,470,110 und US 4,472,712.

U.S. Patent 4,434,421 betrifft ein Verfahren zum Verringern der Anzahl von Kollisionen. Dies erfolgt durch Verringern der Anzahl der Tochterstationen, die Zugang zum Bus erhalten wollen, bis eine Haupt- und eine Tochterstation in Verbindung stehen. Dies unterscheidet sich von der vorliegenden Erfindung darin, daß es sich um ein Rundsendeverfahren handelt, bei dem verschiedene Benutzer die gleichen Nachrichten empfangen können.

US 4,470,110 betrifft ein System zum Austausch von Nachrichten mit einer Schnittstelle. Außerdem wird einem Adressenbit mit einem bestimmten logischen Pegel Priorität erteilt.

Ferner ist von Interesse ein Artikel in IEEE "Automotive Applications of Microprocessors" 1984, Paper No. CH2072-7/84/0000-0083 "A Data Link for Agricultural and Off Highway Communications" von Boyd Nichols, Vijay Dharja und Kanaparty Rao.

Für die vorliegende Erfindung ist es von besonderer Bedeutung, daß sie in der Lage ist, mit einem seriellen Kommunikationsschnittstellenkanal (SCI), einem seriellen peripheren Schnittstellenkanal (SPI) und einem gepufferten seriellen peripheren Schnittstellenkanal (BSPI) zu kommunizieren.

Die Zielsetzung des hier offenbarten seriellen Datenbus-Systems, auch bekannt als Chrysler-Detektoren-Bus (C²D) liegt darin, daß mehrere Mikroprozessoren miteinander über ein gemeinsames Paar Drähte oder einen Bus, der ein Schema ähnlich einer Telefonteilnehmerleitung benutzt, in einfacher Weise kommunizieren können. Alle an den Bus angeschlossenen Mikroprozessoren sind in der Lage, alle auf den Bus übertragenen Nachrichten zu empfangen. Jeder Mikroprozessor, der eine Nachricht über den Bus übertragen will, wartet,

bis der laufende Benutzer beendet ist, bevor die Benutzung versucht wird.

Jedesmal, wenn der Bus frei ist, wird seine Benutzung auf Basis des zuerst Kommens, zuerst Bedienens zugeteilt. Nachdem also eine vorausgegangene Nachricht beendet ist, beginnt irgendein Mikroprozessor seine Nachrichtenübertragung über den Bus und erhält die Benutzung. Wenn jedoch mehrere Mikroprozessoren ihre Nachrichten auf dem Bus genau zur gleichen Zeit übertragen wollen, dann erhält die Nachricht mit der höchsten Priorität die Benutzung des Bus. Alle Nachrichten haben eindeutige Prioritäten und jede Nachricht wird nur von einem Mikroprozessor übertragen.

Eine Zusammenfassung der vorliegenden Erfindung findet sich in US-A-4,706,082 und US-A-4,719,458. Ferner wird hingewiesen auf SAE Information Report "J1567 Collision Detection Serial Data Communications Multiplex" für SAE Multiplexing Committee von Frederick O.R. Miesterfeld am 23.05.1996.

Es wird auf die vorgenannten Patentschriften zur weiteren Erläuterung und Zusammenfassung einiger Grundlagen der hier vorliegenden Erfindung aufmerksam gemacht.

Eine Aufgabe der Erfindung besteht darin, einen SCI Port, einen SPI Port und einen gepufferten SPI Port als Teil einer integrierten Schaltung für eine serielle Datenschnittstelle vorzusehen. Dies erlaubt die Verbindung mit jeder Einrichtung am gleichen Bus, die mit einem dieser drei Ports versehen ist. Der Einbau dieser Ports verbessert die Vereinfachung der seriellen Datenkommunikation, wie sie in den Anmeldungen mit früherem Anmeldetag bezüglich des gleichen Sachverhalts beschrieben ist.

Aufgaben, Merkmale und Vorteile der Erfindung werden aus der folgenden Einzelbeschreibung des bevorzugten Ausführungsbeispiels hervorgeht.

rungsbeispiels, den Ansprüchen und der Zeichnung augenscheinlich. Es zeigen:

- Fig. 1 ein Blockschaltbild eines seriellen Datenbus-Netzwerks;
- Fig. 2 ein Schaltbild des seriellen Datenbus;
- Fig. 3 die Fortsetzung des Schaltbildes für den seriellen Datenbus;
- Fig. 4 ein Blockschaltbild der integrierten Schaltung IC für die Busschnittstelle;
- Fig. 5 eine zeitliche Darstellung eines Beispiels für die Erfassung einer Kollision an der Busschnittstelle IC während des Arbitrationsvorganges;
- Fig. 6 eine Schaltung für den Arbitrationsdetektor 42;
- Fig. 7 des Kollisionsdetektors 44;
- Fig. 8 eine Schaltung des Startbitdetektors 200;
- Fig. 9 eine Schaltung eines Taktfrequenzteilers 201;
- Fig. 10 eine Schaltung eines Wortzählers 202;
- Fig. 11 eine Schaltung eines Flip-Flops 203;
- Fig. 12 eines Bildfehlerdetektors 204;
- Fig. 13 eines Freischaltzählers 206;
- Fig. 14 eine Schaltung eines Freischalt-Flip-Flops 207;
- Fig. 15 eine Schaltung eines digitalen Filters 210;
- Fig. 16 eine Schaltung einer Betriebswahlauswahl 301;
- Fig. 17 eine Schaltung eines SCK-Selektors 302;
- Fig. 18 eine Schaltung eines SCK-Zählers 303;
- Fig. 19 eine Schaltung eines 16-Bit-Puffers und Bit-Umkehrers 304;
- Fig. 20 eines 2- oder 1-Byte-Empfängers 305;
- Fig. 21 eine Schaltung eines 2-Byte-Zählers 306;
- Fig. 22 eine Schaltung eines Start-/Stop-Generators und SPI-Datenpfades 307;
- Fig. 23 eine Schaltung eines SPI-Taktgenerators 308;
- Fig. 24 eine Schaltung der Steuerung 309 für SPI-Übertragung;

- Fig.25 eine Schaltung eines Prüfbetriebsartdetektors 401;
- Fig.26 eine Schaltung einer Rückstellschaltung 402;
- Fig.27 eine Schaltung eines Trenngenerators 403;
- Fig.28 eine Schaltung eines Überbereichsspeichers;
- Fig.29 ein Blockschaltbild der Schaltung für die SPI-Betriebsart einer integrierten Schaltung IC für eine serielle Datenbusschnittstelle;
- Fig.30 ein Blockschaltbild des seriellen Datenbus in einer SCI-Betriebsart;
- Fig.31 ein Blockschaltbild der gepufferten SPI-Betriebsart einer seriellen Datenbusschnittstelle IC;
- Fig.32 ein Flußdiagramm der SPI-Betriebsartverfahren während einer Übertragungsbedingungen;
- Fig.33 ein Flußdiagramm der SPI-Betriebsart im Empfangszustand;
- Fig.34 ein Flußdiagramm der gepufferten SPI-Betriebsart bei Übertragung und
- Fig.35 ein Flußdiagramm der SPI-Betriebsart im Empfang.

Als weitere Dokumente werden genannt US 4,429,384, SAE Technical Paper No. 830536 "Serial Bus Structures for Automotive Applications" von Anthony J. Bozzini and Alex Goldberger vom 28.02.1993; SAE Paper Nr. 860390 "A Small Area Network for Cars" von Ronald L. Mitchell; SAE Paper No. 860390 "In-Vehicle Networking - Serial Data Communication Requirements and Directions" von Frederick H. Phail and David J. Arnett und SAE Paper No. 860389 "Chrysler Collision Detection (C²D) a Revolutionary Vehicle Network" von Frederick O.R. Miesterfeld. Bei diesem System ist der Benutzer-Mikroprozessor an dem Datenbus über eine Busschnittstellenschaltung angeschlossen, die einen Arbitrationsdetektor, einen Puffer (Differentialübertrager), einen Kollisionsdetektor, einen Bustreiber und einen Busempfänger in einer Anordnung aufweist, die eine Differentialübertragung, einen Busfreischaltdetektor, ein Digitalfilter, Takt- und

Synchronisierschaltungen beinhaltet. Die Druckschrift diskutiert auch die Schnittstelle als Standard asynchrone serielle Kommunikationsschnittstelle (SCI) eines Mikrocomputers.

Aufmerksam wird auch auf das vorgenannte U.S. Patent gemacht in Bezug auf eine Teilbeschreibung der in Fig. 1, 2, 3 und 4 dargestellten Schaltungen.

Die Wechselwirkung zwischen dem Arbitrationsdetektor 42, Kollisionsdetektor 44, Wortzähler 202, Wort-Flip-Flop 203, Startbitdetektor 200, Bildfehlerdetektor 204, Freischaltzähler 206, Freischalt-Flip-Flop 207, Taktfrequenzteiler 201, Digitalfilter 210, Bustreiber bestehend aus ODER-Gatter 62 und NAN-Gatter 63 zusammen mit dem Busempfänger 30 in Verbindung mit der Stromquelle 34 und der Stromsenke 36 im Anschluß an den Bus 26 sind beschrieben.

Das Verständnis der vorgenannten Schaltblöcke ist nötig, um die erfindungsgemäßen Verbesserungen zu verstehen. Es wird deshalb auf das U.S. Patent 4,706,082 und US-A-4,719,458 aufmerksam gemacht und die darin enthaltenen Beschreibungen und Zeichnungen.

Die in den Fig. 1 bis 4 dargestellte Verdrahtung für den seriellen Datenbus wird nun erläutert.

SCI MODUS

Die in den Fig. 1 bis 4 dargestellte Schaltung, die für den SCI Modus benutzt wird, wird auch für die anderen Betriebsarten benutzt und ist deshalb Grundlage der gesamten Schnittstelle IC.

Tastet der Startbitdetektor 200 ein gültiges Startbit, so folgt, daß der Wortzähler 202 sich mit der Zeitgabe des

empfangenen Datenwortes synchronisiert. Der Wortzähler 202 erzeugt Impulse, einen für den Kollisionsdetektor 44 und einen anderen für das Wort-Flip-Flop 203. Bei $1/4$ Bitzeit taktet der Wort-Flip-Flop 203 den Arbitrationsdetektor 42. Der Wortzähler 202 löst den Bildfehlerdetektor 204 in der Stopbitzeit aus. Wird kein Stopbit festgestellt, so wird der Freischaltzähler 206 vom Bildfehlerdetektor 204 erweitert bzw. verlängert bis 10 Leerperioden in einer Reihe empfangen sind.

Der Kollisionsdetektor 44 tastet den übertragenen Input und den empfangenen Output. Die Funktion des Kollisionsdetektors 44 liegt darin, Übertragungen zu sperren, die den Bus 26 für Operationen unterbrechen könnten. Wollen zwei oder mehrere Benutzermikroprozessoren 22 etwa gleichzeitig übertragen, so sorgt der Arbitrationsdetektor 42 für die Übertragung auf der Grundlage, wer zuerst kommt, wird zuerst bedient. Versuchen beide Benutzermikroprozessoren 22 synchron zu übertragen, d.h. in einem Arbitrations-Zeitfenster, so läßt der Kollisionsdetektor 44 nur die Fortsetzung der Übertragung mit der höchsten Priorität zu.

Ist ein Benutzermikroprozessor 22 am Bus 26 angeschlossen und übertragungsbereit, so folgt das folgende Verfahren. Zuerst überwacht der Benutzermikroprozessor die IDLE-Leitung und wartet, bis die auf eine logische 0 schaltet, was anzeigt, daß der Bus 26 frei ist. Dann sucht der Benutzer-Mikroprozessor 22 ein 8 Bit-ID Wort zu übertragen, das zu den zu übertragenden Daten gehört. Wenn der Benutzer-Mikroprozessor 22 zuerst die Übertragung begonnen hat oder die höchste Priorität ID besitzt, so lassen der Kollisionsdetektor 44 und der Arbitrationsdetektor 42 die Übertragung zu. Der Mikroprozessor 22 bestätigt die Übertragung, indem er das empfangende ID-Wort liest und es mit dem ID-Wort vergleicht, das zu übertragen gewünscht war. Gibt es eine Bestätigung, daß das gleiche ID-Wort übertragen wurde, so

können dann die Daten übertragen werden. Wenn nicht, so muß der Mikroprozessor prüfen, ob das empfangene ID und die Daten für seinen eigenen Gebrauch bestimmt sind.

Es ist wesentlich, zu erkennen, daß eine Datenkollision infolge äußerer Störungen erfolgen kann oder bei einer Arbitrationsanforderung, wenn lange Datenketten übertragen werden. Der Mikroprozessor 22, der gerade Daten überträgt, kann die übertragenen Daten mit den empfangenen Daten für diesen Typ Kollision vergleichen. Dann sollte der Mikroprozessor 22 passende Maßnahmen ergreifen.

Die Funktion des Freischaltzählers 206 und des Freischalt-Flip-Flops 207 besteht darin, festzustellen, wenn der Bus 26 im Freischaltzustand ist. Dies erfolgt durch Erfassen eines empfangenen Stoppbits. Eine Verzögerung wird dann während einer kurzen Freischaltperiode von 10 Bitzeiten ausgeführt. Der IDLE-Output wird dann auf eine logische Null gesetzt. Die IDLE-Leitung wird beim Empfang eines Startbits auf eine logische 1 gesetzt. Dieses Signal wird auch zur Beendigung der Übertragung benutzt und läßt während einer empfangenen Signalnachricht keine Übertragung zu. Der Freischaltzähler 206 und Freischalt-Flip-Flop 207 liefern auch ein Rückstellsignal an den Kollisionsdetektor 44 am Anfang einer Freischaltperiode.

Eine Anforderung nach Arbitration kann von einem Modul erzeugt werden, das die Übertragung einer langen Datenkette nicht unterbrechen muß. Dies kann dadurch erfolgen, daß der IDLE-Input auf eine logische 0 gesetzt wird; dies erzwingt eine Datenkollision nach dem Übertragen des Startbits aus dem 5. Byte und der übertragende Mikroprozessor 22 muß dies feststellen und die Übertragung anhalten. Er darf aber wiederum sich für den Bus 26 entscheiden, doch darf das Unterbrechungsmodul keine zweite Datenkollision verursachen, wenn es die Arbitration verliert.

Im SCI-Modus sorgt die Busschnittstelle 24 für eine typische, volle, Duplex-asynchrone serielle Kommunikations-schnittstelle zur Übertragung von Datenbytes an und aus dem Mikroprozessor 22.

Zusätzlich zu den asynchronen Datenschnittstellen muß der Mikroprozessor auch eine Schnittstelle für die Busschnittstelle IC 24 der IDLE-Leitung und für Steuerzwecke besitzen.

Wenn im SCI-Modus der Mikroprozessor 22 eine Nachricht an den Bus 26 senden will, dann sendet und empfängt er jedes Byte einzeln und überwacht den Fortschritt. Wenn die Bits eines Bytes von der Übertragungsleitung des SCI-Ports des Mikroprozessors 22 gesendet werden, gehen sie durch die Busschnittstelle IC 24 und auf den Bus 26.

Gleichzeitig werden die Bits am Bus 26 von der Busschnittstelle IC 24 festgestellt und zur Empfangsleitung des SCI-Ports am Mikroprozessor 22 geführt. Dieser vergleicht das empfangene/reflektierte Byte mit dem zuletzt übertragenen Byte und wenn beide gleich sind, erkennt der Mikroprozessor 22, daß das letzte Byte erfolgreich übertragen wurde, so daß er die weiteren Bytes verarbeiten kann. Werden die Nachrichten am Bus 26 von anderen Mikroprozessoren 22 übertragen, so erhält die Busschnittstelle IC 24 dann jedesmal ein Byte über die Empfangsleitung seines SCI-Ports.

Der Mikroprozessor 22 muß die IDLE-Leitung überwachen, um festzustellen, wann das letzte Byte einer Nachricht empfangen worden ist, wann das nächstempfangene Byte die erste Nachricht sein wird und wann es Zeit zur Arbitration ist.

SPI-MODUS

In diesem Modus erzeugt der Wortzähler 202 die Zeitsignale, um den SCK-Zähler 303 zusammen mit dem SCK-Selektor 302 und den Startbitdetektor 200 zu speisen. Der Mikroprozessor 22 wird der Busschnittstelle IC 24 untergeordnet. Soll der Mikroprozessor 22 ein Wort übertragen, so lädt er das Wort in seinen Pufferspeicher und wartet, daß der IDBE-Pin der Busschnittstelle 24 auf eine logische Null geht, also der Bus 26 frei ist.

Der Mikroprozessor pulst dann den CONTROL Pin der Busschnittstelle IC 24. Dies setzt einen IXMIT Enable-Speicher im Programmierer 309 auf Übertragung. Die Busschnittstelle IC 24 wird dann einen Startimpuls erzeugen und beliefert den Mikroprozessor 22 mit 8 SCK-Schiebeimpulse synchron mit dem Startimpuls. Wenn vom Mikroprozessor 22 eine Kollision mit der Nachricht ID festgestellt wird, so stoppt der Mikroprozessor 22 die Übertragung unverzüglich und beginnt mit Empfang. Wird keine Kollision festgestellt, so heißt dies, daß der Mikroprozessor 22 die Entscheidung für sich gewonnen hat und deshalb nun die Datenübertragung vervollständigt.

Der Synchronisierer arbeitet mit dem SPI-Taktgenerator 308 und erzeugt synchronisierte Schiebetaktimpulse zum Empfangen und Übertragen von Daten. Er liefert aber nicht Schiebeimpulse für Start- und Stoppbits. Hier ist eine sehr genaue Taktfrequenz erforderlich, um eine bestimmte Datenübertragungs-/Empfangsgeschwindigkeit zu synchronisieren.

Im SPI-Modus bietet die Schnittstelle IC 24 die Schnittstelleneigenschaften vom SPI-Typ, die an einigen Modell-Mikroprozessoren verfügbar sind.

Bei dem allgemein verfügbaren SPI-Betrieb tauschen zwei Mikroprozessorfamilien, eine als Haupt- und die andere als Unterstation, mit hoher Geschwindigkeit 2 Bytesdaten aus, jeweils ein Bit, wobei die Datensynchronisierung von einem von der Hauptstation gelieferten Taktsignal gesteuert wird.

Ist die Schnittstelle IC 24 im Übertragungsmodus, so wird ein Datenbyte, mit einem Bit jedesmal, gleichzeitig an die Schnittstelle IC 24 übertragen, auf den Bus 26 gegeben, während ein empfangenes reflektiertes Bit aus dem Bus 26 empfangen und zurück zum Mikroprozessor 22 geführt wird. Am Ende eines Übertragungszyklus von 1 Byte hat der Mikroprozessor 22 in seinem SPI-Datenspeicher das Byte, das die Übertragung des Bytes reflektiert, das im SPI-Datenspeicher war, als der Mikroprozessor 22 die CONTROL-Leitung heruntergesetzt hat, um den Beginn des Übertragungszyklus anzufordern.

Beginnt die Schnittstelle IC 24 den Empfang eines Datenbytes aus dem Bus 26, bevor der Mikroprozessor 22 die CONTROL-Leitung heruntersetzt, so setzt die Schnittstelle IC 24 die CONTROL-Leitung herunter und beginnt das SCK-Taktsignal zu erzeugen. So wie jedes Datenbit empfangen wird, wird es aus der Schnittstelle IC 24 heraus in den Mikroprozessor 22 getaktet. Alle Daten im SPI-Datenspeicher des Mikroprozessors vor dem Start des SCK-Signals werden aus dem Datenspeicher nirgendwohin übertragen, wenn die empfangenen Daten in den SPI-Datenspeicher übertragen werden.

In einigen Fällen kann es nützlich sein, die CONTROL-Leitung der Schnittstelle IC 24 mit der SS-Leitung des Mikroprozessors SPI-Ports zu verbinden, alternativ zu anderen Möglichkeiten, die SS-Leitung niedrig zu setzen.

Es gibt eine kleine Zeitverzögerung zwischen der Übertragung des Bits vom Mikroprozessor 22 zur Schnittstelle IC 24 an der Anstiegsflanke eines SCK-Zyklus und der Übertragung des Bits an der Schnittstelle IC zum Mikroprozessor an der Abfallkante des gleichen SCK-Zyklus.

Die Datenübertragungen zwischen dem Mikroprozessor 22 und der Schnittstelle IC 24 erfolgen mit der Bus-Übertragungsgeschwindigkeit 7812,5 Bits/s.

Der SPI-Modus unterscheidet sich vom SCI-Modus hauptsächlich wegen der Unterschiede zwischen den SPI-Typen der Schnittstellen und SCI-Typen der Schnittstellen. Im SPI-Modus hat der Mikroprozessor 22 keinen Zugang zu den am Bus 26 übertragenen Start- und Stoppbits. Im SPI-Modus muß der Mikroprozessor die Bit-Reihenfolge der übertragenen und empfangenen Bytes umkehren.

Der SPI-Modus ist ähnlich dem SCI-Modus insofern, als der Mikroprozessor 22 Daten an und von der Schnittstelle IC 24, jeweils ein Byte, sendet oder empfängt. Beim Übertragen einer Nachricht wird jedes Bit eines übertragenen Bytes gleichzeitig auf den Bus übertragen, während das reflektierte Bit vom Bus 26 empfangen wird.

In SPI ausgedrückt arbeitet der Mikroprozessor 22 im Tochtermodus und die Schnittstelle IC 24 arbeitet als Hauptstelle.

Funktionell ausgedrückt gleicht der Mikroprozessor 22 in der Schnittstelle IC 24, die Schnittstelle IC 24 ist aber mehr dem Mikroprozessor 22 angeglichen, da beide eine Datenübertragung auslösen können. Der Mikroprozessor 22 kann die Übertragung eines Bytes von der Schnittstelle IC 24 durch Niedrigsetzen der CONTROL-Leitung anfordern, aber

dies muß geschehen, bevor die Schnittstelle IC 24 beginnt, Daten vom Bus zu empfangen.

Als Unterstation zur Schnittstelle 24 muß der Mikroprozessor 22 in der Lage sein, die Benutzung des SPI-Ports zu jeder Zeit von Seiten der Schnittstelle IC 24 zu gestatten.

Die Daten werden an den Bus 26 asynchron mit einem Startbit, acht Datenbits und einem Stoppbit übertragen. Die Reihenfolge der Datenbits lautet: Bit mit der kleinsten Wertigkeit (LSB), Bits 1, 2, 3, 4, 5, 6 und MSB. Bei einer SPI-Übertragung überträgt der Mikroprozessor 22 normalerweise das MSB zuerst und das LSB zuletzt, gerade umgekehrt zur Busübertragung.

Um den SPI-Modus zu benutzen, muß der Mikroprozessor 22 die Bitreihenfolge aller übertragenen und empfangenen Bytes umkehren (dieses Problem tritt bei den unten beschriebenen gepufferten SPI-Modus nicht auf).

GEPUFFERTER SPI-MODUS

Der gepufferte SPI-Modus erfordert zusätzliche Schaltkreise im Vergleich zum SCI- und SPI-Modus. Ein 16 Bit-Puffer und Bit-Umkehrer 304 ist zum Empfangen und Übertragen von Daten vorgesehen. Ein Steuer-Flip-Flop bestimmt, ob der Puffer 304 an dem Mikroprozessor 22 oder an die Bus-Übertragungsschaltung angeschlossen ist.

Beim Einschalten wird das Steuer-Flip-Flop mit dem Mikroprozessor 22 verbunden. Dieser ist die Hauptstation und die Schnittstelle IC 24 ist eine periphere Nebenstation. Der Mikroprozessor 22 kann auch an andere periphere ICs angeschlossen werden und die Schnittstelle IC 24 wird von dem CS-Pin (chip select not pin) ausgewählt. Will der Mikroprozessor 22 übertragen, so wählt er die Schnittstelle IC

24, indem er eine "0" am CS-Pin liefert und dann den CONTROL-Pin beobachtet. Geht dieser auf eine logische 1 und signalisiert damit, daß der Pufferspeicher 304 mit empfangenen Daten voll ist und vom Mikroprozessor 22 ausgelesen werden kann, so liefert der Mikroprozessor 22 16 Schiebepulse und liest die Daten zur gleichen Zeit aus, in der er den SPI-Puffer im Block 304 mit den ID und Daten lädt, die er zu übertragen wünscht.

Der Mikroprozessor 22 pulst dann den CONTROL-Pin und die Daten werden zur richtigen Zeit übertragen. Möchte der Mikrocomputer nur ablesen, dann liest er einfach das Bereitstellen der Schiebetaktsignale und pulst den CONTROL Pin nicht. In der Schnittstelle IC 24 ist die Schaltung, um die empfangenen Daten im Pufferspeicher zu halten und um den Empfang neuer Daten zu ignorieren, bis die alten Daten ausgelesen worden sind. Dies stellt sicher, daß die übertragenen Daten überprüft werden können, um sicher zu sein, daß die Arbitration gewonnen wurde; wenn nicht, muß eine Rückübertragung erfolgen.

Im gepufferten SPI-Modus benutzt die Schnittstelle IC 24 ein internes 16 Bit-Schieberegister, genannt 16 Bit-Puffer und Bit-Umkehrer 304, um zwei Datenbytes zwischen dem Mikroprozessor 22 und dem Bus 26 zu puffern, während der typische SPI-Typ der Schnittstelle zur Übertragung der Daten zwischen dem Mikroprozessor 22 und der Schnittstelle IC 24 benutzt wird.

Der Puffer mit zwei Bytes trennt den Mikroprozessor 22 vom Betrieb auf dem Bus 26. Dies ermöglicht, daß der Mikroprozessor 22 sich auf andere Aufgaben mit höherer Priorität konzentriert, wobei mehrere Einrichtungen am SPI-Bus hängen.

Der Mikroprozessor 22 lädt den 2-Byte-Puffer in der Schnittstelle IC 24 mit hoher Geschwindigkeit, benutzt eine SPI-Schnittstelle und signalisiert der Schnittstelle IC 24, die Daten im Puffer zu übertragen.

Die Schnittstelle IC 24 versucht mit Busgeschwindigkeit die gepufferten Daten zum Bus 26 zu übertragen. Während dieses Übertragungsversuches erhält die Schnittstelle IC 24 zwei Bytes reflektierter Daten zurück vom Bus 26, speichert diese im Puffer und sperrt den Puffer vom Empfang weiterer Daten aus dem Bus 26 ab, bis die empfangenen Daten vom Mikroprozessor entladen sind.

Später entlädt der Mikroprozessor 22 wiederum mit Hochgeschwindigkeits-SPI-Transfertechnik die erhaltenen Bytes und lädt gleichzeitig die nächsten Bytes, die übertragen werden sollen.

Während die Schnittstelle zwei Datenbytes zu und von dem Bus überträgt und empfängt, erfolgt über die Schnittstelle IC 24 keine Übertragung von Daten zu und aus dem Mikroprozessor 22 und so muß die Schnittstelle während dieser Zeit nicht vom Mikroprozessor 22 angewählt werden.

Der Mikroprozessor benutzt die IDLE und CONTROL-Leitung, um den Status der Busschnittstelle IC 24 zu tasten und ihren Betrieb zu steuern.

Die hauptsächlichen Unterschiede zwischen dem gepufferten SPI-Modus und dem ungepufferten SPI-Modus bestehen in der Benutzung eines 2-Byte internen Puffers, sowie darin, daß der Mikroprozessor 22 im Modus als Hauptstation und nicht im Modus als Nebenstation arbeitet und daß die Datenübertragung zwischen dem Schnittstellen IC 24-Speicher und dem Mikroprozessor 22 und das Senden/Empfangen von Daten zum/vom Bus 26 getrennt und nicht kombiniert erfolgt.

In den Fig. 2 und 3 ist eine Verdrahtung der Schnittstelle IC erläutert.

Die Modus-Auswahlschaltung 301 ist vorwiegend aus Daten-Multiplexern und Gattern zusammengesetzt, wie sie für ICs bekannt sind. Die Funktion der Modus-Auswahlschaltung 301 besteht darin, den Daten- und Schiebetakt (SCK) Signalfluß in die und aus der Busschnittstelle IC 24 zu steuern. Der MODE- und CS (aktiv niedrig)-Input bestimmt den Modus für die Busschnittstelle IC.

Haben der Input für MODE und CS eine logische 1, so ist die Schnittstelle IC 24 im SCI-Modus. Dann gehen die Daten vom XMIT (Übertragungs)-Pin und werden direkt an einen Output der Modusauswahlschaltung 301 zum Arbitrationsdetektor 42 geschleust. Zusätzlich kommen die zu dem Empfangsdaten (REC)-Pin zu sendenden Daten in die Schaltung 301 vom digitalen Filter 210.

Ist der MODE-Input auf einer logischen 1 und der CS-Input auf einer logischen 0, dann ist die Schnittstelle IC 24 im SPI-Modus. In diesem werden die Daten in die Schnittstelle IC 24 synchron eingegeben, wobei die Schnittstelle 24 die Hauptstation ist. Wenn der Mikroprozessor 22 überträgt, so erzeugt der SCK-Output ansteigende und abfallende Kanten, wobei der Mikroprozessor 22 Daten an der ansteigenden Kante ausgibt und an der abfallenden Kante die Daten bei sich speichert.

Die SCK-Impulse für den SCK-Pin kommen in die Schaltung 301 vom SPI-Taktsignalgenerator 308. Daten, die vom XMIT-Pin kommen, werden zu dem Start/Stoppsbit-Generator und zum SPI-Datenpfad 307 zum Erzeugen von Start- und Stoppsbits geführt. Daten vom Start-/Stoppsbit-Generator 307 kommen in die Modus-Auswahlschaltung 301 zurück und werden dann auf einer Outputleitung zum Arbitrationsdetektor 42 gegeben.

Daten für den REC-Pin von der Modus-Auswahlschaltung 301 kommen während des SPI-Modus vom digitalen Filter 210.

Liegt der MODE-Pin auf einer logischen 0, so ist die Schnittstelle IC 24 im gepufferten SPI-Modus. Dabei dient der CS-Input als echte Chipauswahl.

Ist CS auf einer logischen 0, so ist der REC-Pin im aktiven oder treibenden Zustand. Ist der CS-Input auf einer logischen 1, so ist der REC-Pin im Zustand einer hohen Impedanz und alle in die Schnittstelle IC 24 eintretenden SCK-Impulse werden gesperrt.

In dem gepufferten SPI-Modus ist der Mikroprozessor 22 die Hauptstation, d.h. der Mikroprozessor 22 muß die SCK-Impulse liefern. Der Mikroprozessor 22 wählt den Chip oder die Schnittstelle IC 24 über den CS-Input und erzeugt 16 SCK-Impulse, wobei die zu diesen Impulsen gehörenden Daten in einen 16-Bit-Puffer und Bit-Umkehrer 304 über die Modus-Auswahlschaltung 301 eingegeben werden. Der 16-Bit-Puffer und Bit-Umkehrer 304 wird vom SCK-Selektor 302 getaktet.

Daten vom XMIT-Pin verlassen die Modus-Auswahlschaltung 301 über diesen Pin und gehen in den 16-Bit-Puffer und Bit-Umkehrer 304. Während der Puffer getaktet wird, werden die für den Empfangs-Pin REC vorgesehenen Daten aus dem Puffer über die 2- bzw. 1-Byte-Empfangsschaltung 305 geschaltet.

Die Modus-Auswahlschaltung 301 leistet auch einen Testmodus. Dies ist ein Signal an die Schaltung 301 vom Testmodus-Detektor 401 in Kombination mit der Rückstellschaltung 402. Dieses Signal leitet die Daten aus dem Überbereichsspeicher 61 an den REC-Pin der Modus-Auswahlschaltung 301. Der Testmodus-Detektor 401 besteht im wesentlichen aus zwei D-Flip-Flops und zwei NOR-Gattern.

Der Testmodus-Detektor 401 hat den Zweck, die Modusauswahlschaltung 301 anzusprechen, um Daten aus dem Überbereichsspeicher 61 zum REC-Pin zu leiten, der außerhalb der Modus-Auswahlschaltung 301 liegt. Der Testmodus-Detektor 301 ermöglicht auch, daß der Mikroprozessor 22 eine Rückstellung ausführt.

Die Testmodus-Bedingung wird eingegeben, wenn der A-Input in den Testmodus-Detektor 401 zwei Impulse erhält. Sobald die beiden Pulse gegeben sind, ist der Testmodus bewirkt und die Daten aus dem Überbereichsspeicher 61 werden direkt unter Umgehung des digitalen Filters 201 auf den REC-Pin gegeben.

Der Testmodus wird ausgelöst, wenn der A-Pin noch zweimal gepulst wird. Der Testmodus-Detektor 401 wird auch beim Einschalten zurückgesetzt.

Im Testmodus und mit dem B-Pin auf einer logischen 1 gelangt die Schnittstelle IC 24 in einen Rücksetzzustand. Ist der B-Input Pin auf einer logischen 0, so wird die Schnittstelle IC 24 nicht zurückgesetzt. Die Rücksetzschaltung 402 erzeugt auch ein Zurücksetzen beim Einschalten. Die Rückstellerschaltung 402 besteht im wesentlichen aus zwei NOR-Gattern mit einem internen Rückstellvermögen.

Der Taktsignalteiler 201 bietet dem Mikroprozessor 22 die Fähigkeit einer Teilung um 10, 8, 4 oder 1. Die vier Zustände des Zählers im Taktsignalteil werden vom Input am Pin A und Pin B bestimmt. Der Taktsignalteiler 201 wird ebenfalls von der Rückstellerschaltung 402 zurückgesetzt. Der Taktsignalteiler 201 besteht aus fünf D-Flip-Flops, sechs Gattern und drei Daten-Multiplexern.

Der Arbitrationsdetektor 42 besteht im wesentlichen aus folgenden Gattern: ein D-Flip-Flop und zwei NAND-Gatter.

Der Arbitrationsdetektor arbeitet folgendermaßen: Gewinnt ein Mikroprozessor 22 Zugang zum Bus 26, so wird die IDLE-Leitung der Busschnittstelle IC 24 hochgesetzt und der Mikroprozessor 22 erkennt diesen Zustand und bestimmt, daß der Zugang zum Bus 26 noch möglich ist. Wenn der Mikroprozessor den Bus innerhalb von $1/4$ Bit-Zeiten besetzt, dann wird das Arbitrationsfenster nicht gesetzt und der Mikroprozessor, der den Zugang erhalten will, hat eine Chance, durch die Kollisionsdetektorphase zu gehen. Liefert aber der Mikroprozessor das Startbit auf den Bus 26 nicht innerhalb $1/4$ Bit-Zeiten, dann wird der den Zugang zum Bus versuchende Mikroprozessor 22 vom Bus 26 getrennt, bis der Bus frei ist. Diese Entscheidung wird im $1/4$ Bit-Zeitpunkt vollzogen und von der Bus-Freischaltleitung zurückgenommen.

Der Kollisionsdetektor 44 besteht im wesentlichen aus einem D-Flip-Flop und einem NAND-Gatter. Der Kollisionsdetektor 44 wird in der Mittelbitzeit getaktet. Wird der Detektor 44 getaktet, so bestimmt er, ob der Mikroprozessor 22 eine 1 übertragen hat, während der Bus 26 auf einer logischen 0 ist. Ist dies der Fall, so wird im Kollisionsdetektor 44 ein Speicher gesetzt und die Übertragung von der Schnittstelle IC 24 auf den Bus 26 wird gesperrt. Eine logische 1 am Bus 26 oder am Input zum Kollisionsdetektor 44 beeinflusst den Speicher nicht und der Mikroprozessor 22 hat weiterhin Zugang zum Bus 26. Der Kollisionsdetektor 44 wird im Zustand des freien Bus zurückgesetzt.

Das digitale Filter 210 besitzt drei Flip-Flops und zwei Gatter als Rauschfilter. Zwei D-Flip-Flops sind nach Art eines Schieberegisters verbunden, werden vom Taktgenerator des Systems getaktet und die Q-Outputs der Flip-Flops führen zu einem AND-Gatter. Auch die Q-Outputs gehen in ein NAND-Gatter. Das erste NAND-Gatter geht in den Setzeingang eines RS-Flip-Flops und das zweite Flip-Flop geht in den Rücksetzeingang des RS-Flip-Flops. Damit ergibt sich ein 2

aus 3 Detektor. Das digitale Filter 210 erhält seinen Input aus dem Überbereichsspeicher 61 und liefert seine Daten. Das digitale Filter 210 wird bei einer Rückstellbedingung in Rückstellung gehalten.

Der Wortzähler 202 besteht basismäßig aus 12 B-Flip-Flops, die als Asynchrone Zähler aufgebaut sind. Der Zweck des Wortzählers 202 liegt darin, das Zählen zu starten, wenn ein Startbit aus dem Startbit-Detektor 200 kommt. Der Wortzähler 202 liefert dann die Zeitgabe für den Kollisionsdetektor 44, den Arbitrationsdetektor 42, den Bildfehlerdetektor 204 und andere SPI-Zeitfunktionen.

Der Wortzähler 202 liefert das $1/4$ Bitzeit-Taktsignal für den Wortzähler 202, dessen Output von dem 6. Flip-Flop in der Zählerkette aufgenommen wird.

Der Kollisionsdetektor 24 erhält sein $1/2$ Bitzeitsignal vom 7. Flip-Flop in der Zählerkette. Auch wird ein Signal aus der Zählerkette decodiert, um ein Signal in der Mitte des 10. Bits (Stoppbit) zu liefern. Dieses Signal aus dem Wortzähler 202 wird dazu benutzt, die Bilddetektorschaltung 204 zu takten. Ein Taktzyklus-Verzögerungssignal aus dem Wortzähler 202 wird dazu benutzt, im Block 203 das Wort-Flip-Flop zu setzen. Die SPI-Steuer-Outputs vom Wortzähler im Block 202 werden vollständig in den jeweiligen Abschnitten der Schnittstelle IC 24 im Rest der Fig. 2 und 3 abgedeckt.

Die Funktion des Wort-Flip-Flops im Block 203 besteht darin, nach der Mitte des 10. Bits oder nach einer Rücksetzbedingung auch ein Startbit aus dem Startbit-Detektor 200 zu sperren, wenn das Startbit für die Zeit von $1/4$ Bit vorhanden war.

Ist die Zeit des Startbits, nach der Mitte des 10. Bits, oder nach dem Rücksetzen kürzer als $1/4$ Bitzeit, dann

sperrt das Wort-Flip-Flop 203 nicht auf ein Startbit hin und der Wortzähler 202 kann sich über den Startbit-Detektor 200 zurückstellen. Wenn jedoch das Startbit länger als $1/4$ Bitzeit vorhanden gewesen ist, so wird das Wort-Flip-Flop 203 gesperrt und nicht zurückgesetzt bis zur Mitte des 10. Bits (ein Stoppbit). Das Rücksetzen des Wort-Flip-Flops im Block 203 erfolgt über ein Ausgangssignal des Wortzählers 202.

Der Startbit-Detektor 200 besteht hauptsächlich aus einem NOR-Gatter und arbeitet in Verbindung mit dem Wort-Flip-Flop 203.

Diese beiden Schaltungen arbeiten zusammen, nachdem die Mitte des 10. Bits vorbei ist oder gerade nach einem Zurücksetzen. Die Schaltungen warten auf ein Startbit aus dem Bildfehler-Detektor 204 und wenn dieses Startbit auftritt, so wird der Wortzähler 202 über den Startbit-Detektor 200 eingeschaltet.

Bleibt das Startbit stehen, so bleibt der Wortzähler 202 eingeschaltet, und, falls das Startbit während $1/4$ Bitzeit ansteht, wird das Wort-Flip-Flop 203 gelatched und somit wird auch der Wortzähler 202 über den Startbit-Detektor 200 eingeschaltet.

Der Startbit-Detektor 200 ist ein NOR-Gatter mit Eingängen vom Bildfehler-Detektor 204 und dem Wort-Flip-Flop 203. Der Ausgang des Startbit-Detektor 200 geht zum Rücksetzeingang des Wortzählers 202, so daß alle beide Schaltungen den Wortzähler 202 einschalten können. Beträgt das Startbit weniger als $1/4$ Bitzeit, dann schaltet der Startbit-Detektor 200 den Wortzähler 202 aus. Beobachtet der Startbit-Detektor 200 ein Startbit, so wird der Ausgang des Freischalt-Flip-Flops 307 aktiviert und damit IDLE hochpegelig gemacht. Der Bildfehler-Detektor 204 erhält Daten aus dem

digitalen Filter 210 und leitet seine Daten auf den Startbit-Detektor 200. Der Bildfehler-Detektor 204 wird in der Mitte des 10. Bits oder des Stoppbits getaktet. Ist das Stoppbit eine logische 1, dann ist es gültig und die Datendurchleitung durch die Schaltung wird fortgesetzt. Wenn aber das 10. Bit eine logische 0 ist, dann bedeutet dies eine Bildfehler-Bedingung und der Bildfehler-Detektor 204 sperrt jedes weitere Startbit am Eintritt in den Startbit-Detektor 200. Die Wirkung ist die, daß der Wortzähler 202 in einer Rücksetzbedingung gehalten wird und der Freischaltzähler 206 läuft, bis der Bus während mindestens 10 Bitzeiten frei geworden ist. Der Bildfehler-Detektor 204 wird bei freiem Bus zurückgesetzt.

Der Freischaltzähler 206 soll die Bitzeiten zählen, nachdem ein Wort vervollständigt worden ist. Der Zähler zählt aufwärts bis zu 10 Bitzeiten und setzt dann das Freischalt-Flip-Flop 207 und den Bildfehler-Detektor 204 zurück. Wenn während des Auszählens der Freischaltzeiten auf dem Bus eine 0 von weniger als $1/4$ Bitzeit auftritt, werden die oberen vier Bits des Freischaltzählers 206 zurückgesetzt und die 10 Bitzeiten werden verlängert. Die Busdaten vom digitalen Filter 210 werden bei $1/2$ Bitzeitdauern getastet, um den oberen vier Bits im Freischaltzähler 206 eine gewisse Geräuschempfindlichkeit zu verleihen. Deshalb müßte der kleinere als der $1/4$ Bitzeit-Nullwert auf dem Bus 26 während des $1/2$ Bitzeitfensters auftreten, um die oberen vier Bits im Freischaltzähler 206 zurückzustellen und damit die Freischaltzeit zu verlängern.

Das Freischalt-Flip-Flop 207 besteht aus einem Flip-Flop, einem AND-/NAND-Gatter und einem Transistor mit einem aktiven Pull-up. Die Aufgabe ist, den Bus 26 als besetzt zu melden, wenn über den Bus 26 Verkehr läuft. Das Freischalt-Flip-Flop 207 gerät nach einem Rücksetzen beim Einschalten in den Setzzusstand. Dieser wird dann über ein AND-Gatter

geleitet. Der Ausgang des AND-Gatters steuert einen Transistor. Wenn also das AND-Gatter hochgesetzt ist, so ist der LED-Pin niedrig und umgekehrt. Der andere Eingang des AND-Gatters kommt aus dem Startbit-Detektor 200.

In der Annahme, daß in der Schnittstelle IC 24 Zustände herrschen, wie zurückgesetzt gerade nach einem Einschalten oder nach einer langen freien Periode, größer als 10 Bitzeiten, geht der IDLE-Pin auf niedrig. Sobald am Bus 26 eine logische 0 festgestellt wird, geht der Ausgang des Startbit-Detektors 200 auf niedrig und signalisiert dem AND-Gatter, den IDLE-Pin hochzusetzen. Falls dann das Startbit durch Rauschen deformiert oder durch Rauschen erzeugt ist, und weniger als $1/4$ Bitzeit beträgt, so wird der Ausgang des Startbit-Detektors 200 hochgesetzt und der IDLE-Output kehrt in den Niedrigzustand zurück. Dauert jedoch das Startbit mehr als $1/4$ Bitzeiten, so ist es wahrscheinlich ein gültiges Startbit und das Wort-Flip-Flop 203 wird latchen. Dies setzt den Freischalt-Flip-Flop 207 zurück und wenn der Ausgang des Freischalt-Flip-Flops 207 an das AND-Gatter geführt ist, so wird sichergestellt, daß der Ausgang des AND-Gatters eine logische 0 ist, womit der IDLE-Pin hochgesetzt wird. Der Bus 26 meldet dann, besetzt zu sein, bis ein Signal aus dem Freischaltzähler 206 das Freischalt-Flip-Flop setzt, so daß damit der IDLE-Pin zurück auf niedrig gesetzt wird und einen freien Bus anzeigt. Der Unterbrecher-Generator 403 soll es einem Mikroprozessor 22 gestatten, einen Nullzustand am Bus 26 zu erzwingen. Dieser Nullzustand am Bus 26 kann nur dann dem Bus 26 aufgezwungen werden, nachdem ein Mikroprozessor mindestens 4 Bytes übertragen hat. Beim ersten Datenbit des 5. Bytes wird die Unterbrechung aktiviert. Geht dann der Mikroprozessor 22 auf die Freischaltleitung zurück, so wird der Nullzustand dem Bus 26 erteilt. Wird der Freizustand verlassen, so hat der Unterbrechergenerator 403 keinen Effekt

mehr auf den Bus 26. Der Unterbrechergenerator 403 besteht im wesentlichen aus drei D-Flip-Flops und vier Gattern.

Die SPI-Übertrager-Planungs- und Steuerschaltung im Block 309 besteht aus etwa drei D-Flip-Flops, neun Gattern, einem Datenmultiplexer und einem Transistor mit einem aktiven Pull-up. Die Übertrager-, Planungs- und Steuerschaltung im Block 309 wird im SPI-Modus und gepufferten SPI-Modus verwendet. Ihre hauptsächliche Funktion besteht in einer Steuerungsfunktion, wenn die Daten aus einem Mikroprozessor 22 auf den Bus 26 geführt werden. Will der Mikroprozessor 22 Daten übertragen, so geht er niedrig auf der CONTROL-Leitung. Im ungepufferten SPI-Modus, nach Niedersätzen auf der CONTROL-Leitung wird die SPI-Übertrager-, Planungs- und Steuereinheit im Block 309 von der Busschnittstelle IC 24 auf niedrig geschaltet. Wenn die CONTROL-Leitung, die an den Block 309 angeschlossen ist, unmittelbar nach dem Niedrigsetzen der IDLE-Leitung niedrig gesetzt worden ist, wird eine 2 Bit-Zeitverzögerung eingesetzt, bevor ein Startbit auf den Bus 26 ausgegeben wird. Ein Signal aus dem Freischaltzähler 206 bestimmt 1,5 Bitzeiten der 2-Bit-Zeitverzögerung. Sobald nach dem Freischalten die 1,5 Bitzeit erreicht worden ist, setzt das Signal vom Freischaltzähler 206 einen Flip-Flop in der Planungs- und Steuerschaltung 309. Dieses Flip-Flop wird von einem Signal aus dem Freischalt-Flip-Flop im Block 207 in der Freischaltzeit zurückgesetzt.

Wenn deshalb einmal dieses Signal gesetzt ist und der Steuerpin auf niedrig steht, so kann ein anderes Flip-Flop in der Planungs- und Steuerschaltung 309 1/2 Bitzeiten später gesetzt werden. Dieses Flip-Flop wird von einem anderen Signal aus dem Freischaltzähler 206 getaktet. Ist einmal dieses Flip-Flop im Block 309 gesetzt, so wird sein Ausgangssignal an den Start/Stopp-Generator und den SPI-Datenpfadblock in 307 geführt, wo ein Startbit erzeugt wird. Der

Ausgang des Flip-Flops wird mit einem Signal aus dem Wortzähler 202 am Ende der Startbitzeit zurückgesetzt.

Wenn ein Startbit auf den Bus 26 geführt wird und der Mikroprozessor 22 die CONTROL-Leitung nicht auf niedrig gesetzt hat, dann taktet ein anderes Signal aus dem Wortzähler 202 den Steuerspeicher im Block 309 und schaltet die CONTROL-Leitung auf niedrig um. Dies ereignet sich am Ende eines Startbits.

Zwei Eingänge, einer vom Arbitrationsdetektor 42 und ein anderer vom Kollisionsdetektor 44, teilen dem Planungs- und Steuergerät im Block 309 mit, ob es Kollisionen oder verlorene Arbitrationen eingegeben hat, so daß dann keine weiteren Startbits mehr von der Schnittstelle IC 24 erzeugt werden, bis der Bus wieder frei ist.

Die Planungs- und Steuerschaltung im Block 309 im gepufferten SPI-Modus arbeitet im wesentlichen in der gleichen Weise wie im oben beschriebenen ungepufferten SPI-Modus, mit nur wenigen kleineren Abweichungen. Will der gepufferte SPI-Mikroprozessor eine Übertragung ausführen, so muß er zuerst sein 16 Bitpuffer im Block 304 mit Daten laden. Dann muß er die CONTROL-Leitung niedrig setzen und diese wird dann von der Schnittstelle IC 24 auf niedrig gesetzt.

Der Input für die Planungs- und Steuerschaltung 309 vom SCK-Zähler 303 teilt der Schaltung 309 mit, daß, sobald 16 Bits gelesen worden sind und 1 Byte aus dem Datenbus 26 ankommt, die CONTROL-Leitung auf niedrig gesetzt wird, wenn das Eingangssignal aus dem Wortzähler 202 den Steuerspeicher taktet. Dies ereignet sich am Ende der Startbitzeit. Die Bestimmung, ob die 16 Bits gelesen worden sind oder nicht, erfolgt durch ein anderes Signal aus dem SCK-Zähler 303 für die Planungs- und Steuerschaltung 309. Sind 16 Bits

nicht gelesen worden, so wird die CONTROL-Leitung nicht niedrig gesetzt.

Nimmt man an, daß die 16 Bits gelesen worden sind und der Mikroprozessor sich auf die CONTROL-Leitung schaltet, dann spricht die Schnittstelle IC 24 genauso an wie im ungepufferten SPI-Modus, indem sie ein Startbit erzeugt, acht Datenbits auf den Bus taktet, worauf ein Stoppbit folgt, ein Startbit, dann acht weitere Datenbits, gefolgt von dem Stoppbit. Im gepufferten SPI-Modus kehrt die CONTROL-Leitung auf hoch am Ende des 9. Bits im zweiten Byte zurück. Dies wird über ein Signal vom SPI-Taktsignalgenerator 308 vollzogen, das dem Steuergerät 309 zugeführt wird, das den Steuerspeicher in der Schaltung 309 taktet.

Im gepufferten SPI-Modus wird die CONTROL-Leitung wieder hoch gesetzt am Ende der 9. Bitzeit. Es wird durch Takten des Steuerspeichers über ein Signal vom Start/Stoppbit-Generator im SPI-Datenpfadblock 307 ausgeführt.

Der SPI-Taktsignalgenerator im Block 308 besteht aus einem Gatter und einem RS-Flip-Flop und wird benutzt, die SCK-Impulse für den Mikroprozessor 22 und den 16 Bit Pufferbit-Umkehrer im Block 304 zu erzeugen.

Diese Impulse sind so beschaffen, daß eine Anstiegsflanke zu Beginn des zweiten Bits erfolgt, welches das erste Datenbit ist und das eine Abfallflanke in der Mitte des zweiten Bits auftritt. Dies dauert für acht Bitzeiten. Dies wird nun in der Mitte des 9. Bits der Fall sein.

Die Hauptzeitbasis für das SCK-Signal stammt aus dem Wortzähler 202 wie für den SPI-Taktgenerator 308 dargestellt. Dies ist $1/2$ Bitzeit-Taktsignal.

Der SCK-Output aus diesem Block wird von dem 1/2 Bitzeit-Taktsignal abgeleitet, aber der SPI-Taktsignalgenerator 308 muß alle Taktimpulse vor dem Ende des Startbits sperren. Der SCK-Output des SPI-Taktsignalgenerators 308 wird auf die Modus-Auswahlschaltung 301 und den SCK-Selektor im Block 302 geführt.

Ein Signal vom Wortzähler 202 wird dem SPI-Taktsignalgenerator 308 zugeführt und wird am Ende eines Startbits aktiviert. Dieses Signal wiederum löscht ein internes Flip-Flop im SPI-Taktsignalgenerator 308, dessen Output für den Steuerspeicher zurückgesetzt wird. Wenn einmal dieses Signal gelöscht ist, beginnt der SCK-Generator-Signalausgang vom SPI-Taktsignalgenerator 308 in der bereits beschriebenen Weise zu takten. Dieses Signal wird dann in der Mitte des 9. Bits gesperrt. Dies erfolgt durch ein anderes Signal vom Wortzähler 202, das den SCK sperrt. Dieses Signal setzt den internen Flip-Flop des SPI-Taktsignalgenerators 308. Wie vorher stellt dieser Ausgang das Rücksetz-Steuerspeichersignal dar, das an die Planungs- und Steuerschaltung 309 geführt wird und das beim Einschalten zurückgesetzt wird.

Betrachtet man nun den SPI-Start/Stoppsbit-Generator und den Datenpfadblock in 307, so ist dies die Schaltung für einen Datenpfad im gepufferten und ungepufferten SPI-Modus. Der Block besteht im wesentlichen aus einem RS-Flip-Flop und zwei Gattern. Für gewöhnlich gehen die Daten frei vom Input zum Block 307 vom Modus-Auswahlblock 301 zum Ausgang des Blockes 307 zurück in die Modus-Auswahlschaltung 301. Die Daten werden aber gestört, wenn ein Startbit oder ein Stoppsbit erzeugt wird. Kommt die Busschnittstelle IC 24 aus einem Zurücksetzen beim Einschalten, so blockiert das Signal des Arbitrationsdetektors 42 die Daten am Bus 26. Das heißt, ein Stoppegel am Bus 26 wird erzwungen. Ist ein Mikroprozessor 22 im SPI-Modus und bereit zur Übertragung,

so veranlaßt das Signal von der Planungs- und Steuerschaltung 309 an die Datenpfadschaltung in 307 ein Startbit am Bus 26. Am Ende des Startbits setzt das Eingangssignal vom Wortzähler 202 an den Datenpfadblock 307 ein internes Flip-Flop im Block 307, dessen Ausgang auf den SPI-Taktsignalgenerator 308 und den Unterbrechergenerator 403 geführt wird, so daß gültige SPI-Daten auf den Bus 26 gelangen können. Die Daten können dann frei verkehren, bis das Eingangssignal vom Wortzähler 202 das Ausgangssignal des Blockes 307 setzt, das den SPI-Taktsignalgenerator 308 und der Planungs- und Steuerschaltung 309 zugeführt wird. Dies ruft dann ein Stoppbit am Bus 26 hervor und blockiert alle weitere SPI-Daten. Dieses Signal vom Wortzähler 202 wird auch bezeichnet als "Stoppbitzeit-Setzen".

Der SCK-Selektor im Block 302 wird nur in dem gepufferten SPI-Modus benutzt. Er besteht aus einem Flip-Flop, einem Datenmultiplexer und einem Gatter. Der SCK-Selektor 302 hat eine Funktion, die sich beschreiben läßt, welche Quelle den 16 Bit-Puffer und Bit-Umkehrer 304 taktet. Das Takten erfolgt über den Output des SCK-Selektors 302 in dem 16 Bit-Puffer und Bit-Umkehrer 304. Das Takten kann aus einer von zwei Quellen stammen, entweder dem intern erzeugten SCK-Signal vom SPI-Taktsignalgenerator 308 oder dem Mikroprozessor 22, wobei das Signal aus dem Modus-Auswahlblock 301 in den SCK-Selektor 302 gelangt.

Kommt die Busschnittstelle IC 24 aus einem Rücksetzzustand beim Einschalten, wie von der Rücksetzschaltung 402 angezeigt, oder nach einem 2-Byte-Empfangssignal, wie von dem 2-Byte-Zähler 306 angezeigt, oder nach einem freien Bus, wie von dem Freischalt-Flip-Flop 207 angezeigt, so kann der 16-Bit-Puffer und Bit-Umkehrer 304 vom Mikroprozessor 22 getaktet werden. Wenn der Mikroprozessor den 16 Bit-Puffer und Bit-Umkehrer 304 taktet, nämlich 16 x am SCK-Pin, so ist der Mikroprozessor nun in der Lage, neue Daten zu über-

tragen oder vom Bus 26 zu empfangen. Sobald die 16 Bits eingegangen sind, wird der Ausgang des SCK-Zählers 303 auf hoch gesetzt am Ende des ersten Startbits, wegen des Signals vom Wortzähler 202, der bei der ersten Bitzeit zurückgesetzt wird.

Wenn der Ausgang des SCK-Zählers 303 dem SCK-Selektor 302 zugeführt wird, so bestimmt dies, daß der 16 Bit-Puffer und Bit-Umkehrer 304 über das Signal MUXCNTRL getaktet wird. Wenn das MUXCNTRL-Signal aus 302 zurückgesetzt wird, so wird der 16 Bit-Puffer und Bit-Umkehrer 304 von der Schnittstelle IC 24 getaktet. Ist es aber gesetzt, so taktet der Mikroprozessor 22 die Schaltung 304. Der MODE-Input in die Schaltung 304 setzt das zwischen einem der Ausgänge des SCK-Zählers 303 und einem der Eingänge des 16 Bit-Puffers und Bit-Umkehrers 304 geteilte Signal in in einen logischen Null-Zustand, so daß der 16 Bit-Puffer und Bit-Umkehrer 304 immer vom Mikroprozessor 22 getaktet wird.

Der 16 Bit-Puffer und Bit-Umkehrer im Block 304 besteht aus einem Schieberegister mit 16 Flip-Flops und 16 Data-Multiplexern. Er ist als Schieberegister geschaltet. 2 zu 1 Multiplexer-Gatter dienen zum Ändern des Rückföhrpfades. Wenn ein Signal von der multiplexen Steuerleitung (MUXCNTRL) des SCK-Selektors 302 auf einer logischen 0 ist, so werden die Daten im 16 Bit-Puffer und Bit-Umkehrer 304 von rechts nach links verschoben. Daten aus dem Mikroprozessor 22 werden dem ersten Flip-Flop in der Kette der mit "Daten für BSPI" bezeichneten Signalleitung eingegeben.

Dann werden Daten das letzte Flip-Flop in Richtung REC-Pin verlassen.

Ist das MUXCNTRL-Signal aus dem Block 302 eine logische 1, so gelangen die Daten aus dem digitalen Filter 210 in das 8. Flip-Flop. Die Daten werden dann vom 8. Bit in Richtung

zum ersten Bit verschoben. Das erste Bit leitet dann Daten zum 16. Bit und die Daten vom 16. Bit werden in Richtung auf das 9. Bit hin nach unten verschoben. Die Daten werden aus dem 9. Flip-Flop auf den Datenbus 26 geführt.

Der SCK-Zähler besteht aus fünf D-Flip-Flops, die als asynchroner Zähler zusammengeschaltet sind sowie aus Flip-Flops und Gatter zur Steuerung des Zählers. Der Zweck des SCK-Zählers liegt darin, die SCK-Impulse zu zählen, die vom Mikroprozessor 22 kommen. Damit werden die Impulse von dem Signal gezählt, das von der Modus-Auswahlschaltung 301 kommt.

Nach dem Einschalten wird der SCK-Zähler 303 zurückgesetzt. Während der Mikroprozessor 22 zählt, zählt der fünfstufige, asynchrone Zähler die Anzahl der eintreffenden SCK-Impulse. Ist die Zahl 16 erreicht, so wird der letzte Zähler in der Stufe gelatched und sein Ausgang geht auf eine logische 1 und zur Planungs- und Steuerschaltung im Block 309.

Sobald der Mikroprozessor 22 zu senden beginnt und das Inputsignal NQSFF8 von 0 auf 1 umschaltet, so setzt auch dies den SCK-Zähler zurück. Ein Ausgang vom SCK-Zähler 303 setzt das Signal NQSFF8 zurück; es schaltet auch den Steuerpin niedrig während des gepufferten SPI-Modus, wenn ein Startbit vom Bus 26 eintrifft, wenn der Mikroprozessor aus dem Bus 26 16 Bits gelesen hat. Die 2- oder 1-Byte-Empfängerschaltung in 305 ist basismäßig ein RS-Flip-Flop und ein Daten-Multiplexer. Seine Funktion liegt darin, zwischen einem 2-Byte-Empfangssignal und einem 1-Byte-Empfangssignal zu unterscheiden. Dies deshalb, weil typischerweise die meisten Nachrichten mindestens mit 2 Bytes erfolgen. Bejahendenfalls meldet sich der Mikroprozessor 22 und empfängt die 2 Bytes und die Busschnittstelle IC 24 wird ihren 16 Bit-Puffer und Bit-Umkehrer 24 voll haben und das erste Byte wird zuerst und das zweite Byte als zweites ausge-

taktet. Wenn aber die Schnittstelle IC 24 nur eine 1-Byte-Nachricht empfängt, dann sitzt nur ein einziges gutes Byte im Block 304. Dann passiert folgendes: Das erste aus dem 16 Bit-Puffer und Bit-Umkehrer 304 ausgetaktete Byte wird das schlechte Byte sein und das nächste Byte wird das empfangene Byte sein. Um dies zu korrigieren, fügten wir eine Schaltung hinzu, um festzustellen, ob 1 oder 2 Bytes empfangen werden. Wenn die Übertragung beginnt, wird der Block 305 von einem Ausgangssignal des SCK-Zählers 303 zurückgesetzt, der einen fehlerhaften 1-Byte-Empfang erzeugt. Wenn nach der Übertragung zwei Bytes empfangen worden sind, wird der Block von dem 2-Byte-Zähler im Block 306 gesetzt.

Wenn es aber einen 1-Byte-Empfang gab, dann werden die Daten zum Empfangspin aus der Mitte des 16 Bit-Puffers und Umkehrers 304 übertragen. Gab es einen 2-Byte-Empfang, dann werden die Daten zum Empfangspin vom Ende des Puffers im Block 304 übertragen.

Der 2-Byte-Zähler 304 besteht im wesentlichen aus zwei Flip-Flops und einem NOR-Gatter zum Zählen der Bytes. Die Funktion des Blockes besteht darin, bis zu zwei Empfangsbytes aufzuzählen. Der 2-Byte-Zähler wird über den Output vom Arbitrationsdetektor 42 getaktet. Das Takten erfolgt in der Mitte des 9. Bits, welches das 8. Datenbit ist.

Der Zähler dient zwei Zwecken, wenn er bis 2 hinaufzählt; er wird den SCK-Selektor 302, der von der Schnittstelle IC 24 getaktet wird, auf das Takten vom Mikroprozessor 22 umschalten. Dies erfolgt über den Output, der den 2-Byte-Zähler 302 mit dem 2- oder 1-Byte-Empfangsblock in 305 verbindet. Das gleiche Signal setzt auch das Flip-Flop im Block 305 nach einem 1-Byte-Empfang. Der Überbereichsdetektor im Block 60 stellt fest, wann der Bus 26 3,13 V überschreitet oder unter 1,8 V absinkt. Dies wird als Außerbereichs-Zustand erkannt und das Ausgangssignal dieses Blocks wird

hoch. Sind die Eingangsspannungen für den Block unterhalb 3,13 V und über 1,8 V, dann gibt es keine Außerbereichs-Bedingung und der Blockausgang ist niedrig. Der Block ist im wesentlichen ein Detektor, der in der I/O-Schaltung eingebaut ist.

Die Überbereichssperre (ORL) im Block 61 ist im wesentlichen ein D-latch. Wird ein Überbereichs-Zustand im Block 60 festgestellt, so spricht der Überbereichs-Detektor 60 die Überbereichs-Sperre im Block 61 an, die dann nach dem letzten gültigen Teil der Daten vor dem Eintritt des Überbereichs-Zustandes sperrt. Gibt es keine Überbereichs-Bedingung, dann gelangen die Daten frei aus dem Block 60 durch den Block 61 und hinaus zum digitalen Filter 210.

Das 3-Input-OR-Gatter gemäß Block 62 wird in Verbindung mit dem Kollisionsdetektor 44 und dem Arbitrationsdetektor 42 und dem multiplexten Output der Modus-Auswahlschaltung 301 benutzt, der die übertragenen Daten führt. Ist jeder der beiden Detektoren gesetzt, so ist der Ausgang des Gatters ein konstanter 1 Wert. Nach Durchleiten durch die Schaltung 63 erzwingt dies eine 0. Damit werden die Stromquellen im Block 34 und 36 nicht eingeschaltet und damit wird die Busschnittstelle IC 24 nicht auf den Bus 26 übertragen. Wenn die Detektoren 44 und 42 nicht gesetzt sind, so gelangen die Daten vom multiplexten Output der Modus-Auswahlschaltung 301 frei in das NAND-Gatter im Block 63 und werden auf den Bus 26 übertragen.

Das 2-Input-NAND-Gatter gemäß Block 63 wird Daten entweder vom Block 62 oder vom Unterbrecher-Generator im Block 403 übertragen. Wenn jeder Input eine logische 0 ist, so ist der Output des NAND eine logische 1 und schaltet damit die Stromquellen in den Blöcken 34 und 36 an.

Die Bus-Plus-Stromquelle im Block 34 wird abgeschaltet und hat keinen Effekt auf die Bus-Plus-Leitung, wenn der Eingang in den Block eine logische 0 ist. Ist aber der Eingang in den Block eine logische 1, so wird die Stromquelle 34 angeschaltet. Ist dies erfolgt, so fließt Strom von V_{CC} zur Bus-Plus-Leitung.

Die Bus-Minus-Stromquelle im Block 36 wird abgeschaltet und hat keinen Effekt auf die Bus-Minus-Leitung, wenn der Eingang zum Block eine logische Null ist. Ist aber der Eingang des Blocks eine logische 1, so wird die Stromquelle eingeschaltet und der Strom fließt aus der Bus-Minus-Leitung zu Masse.

Fig. 5 erläutert ein Beispiel einer Kollisionsfeststellung in der Busschnittstelle IC mit einem CONTROL-Signal.

Fig. 6 bis 28 sind Logikschaltungen, die in den Blockschaltbildern Verwendung finden.

In den Fig. 29 und 30 ist die Verdrahtung der seriellen Datenbus-Schnittstelle IC 24 in zweifacher Weise dargestellt. In Fig. 30 ist die Busschnittstelle IC 24 im seriellen Kommunikationsschnittstellen-Modus (SCI) dargestellt. Das heißt, es wird nur die Verdrahtung dargestellt, die einzig und allein für SCI benutzt wird und die gemeinsam ist mit jedem anderen Modus im Betrieb mit dem seriellen Datenbus, nämlich SPI und gepufferte SPI.

Fig. 29 vervollständigt die Darstellung in Fig. 30 durch Hinzufügen von Bausteinen aus der Schnittstelle IC 24, die in einem SPI-Modus (in einem ungepufferten Zustand) benutzt wird.

Das Blockschaltbild in Fig. 30 ist in dem eingangs genannten Schrifttum dargestellt und dies soll hier nicht wiederholt werden.

Die in Fig. 29 dargestellte Schaltung für den ungepufferten SPI-Modus vervollständigt nur die Schaltung der Fig. 30 mit dem Start/Stopp-SCK-Generator 307 und den SCK- und CONTROL-Leitungen. Auch ist diese Figur mit dem Freischalt-Steuerdetektor 54 ergänzt, um zu zeigen, daß dieser Detektor aus dem Freischaltzähler 206, dem Freischalt-Flip-Flop 207 und der Planungs- und Steuerschaltung 309 besteht.

Die in Fig. 31 dargestellte Schaltung für den gepufferten SPI-Modus ist ferner gegenüber den Schaltungen in Fig. 29 und 30 um einige Schaltungen und Leitungen ergänzt, die in anderen anhängigen Patentanmeldungen erläutert sind, nämlich um die Puffer-Zugangssperre 302, die man auch als SCK-Selektor bezeichnen kann, und den Puffer 304, der auch als 16 Bit-Puffer und Bit-Umkehrer bezeichnet ist. Ferner sollen die Übertrager/Empfangs-Leitungen am Puffer 304 beachtet werden.

In den Fig. 32 und 33 soll das SPI-Verfahren im Zustand der Übertragung und des Empfangs beschrieben werden.

Im SPI-Übertragungszustand soll die Schnittstelle IC 24 einen SPI-Modus zur Datenübertragungssteuerung benutzen. Der Modus beginnt im Block 600 und die Schnittstelle IC 24 prüft im Entscheidungsblock 602, ob die IDLE-Leitung niedrig ist. Ist sie nicht niedrig, so wartet die Schnittstelle 24 ab, bis sie es ist und rückt dann zum Entscheidungsblock 604 vor, um die CONTROL-Leitung zu prüfen, ob diese niedrig gesetzt worden ist. Wenn nicht, wartet die Schnittstelle IC, bis sie niedrig gesetzt ist und schreitet dann zum Block 606, um die CONTROL-Leitung in der Niedrigbedingung zu latches.

Dann prüft die Schnittstelle IC 24 im Entscheidungsblock 608, ob die IDLE-Leitung zwei Bitzeiten lang niedrig gewesen ist. Wenn nicht, wird auf das Eintreten dieser Bedingung gewartet und dann folgt der Entscheidungsblock 610, um ein Startbit auf den Bus 26 zu geben und die IDLE-Leitung auf eine logische 1 zu setzen.

Dann prüft die Schnittstelle IC im Block 612 das Ende eines Startbits und wartet, bis das Ende eines Startbits auftritt, worauf Block 614 folgt, um den Mikroprozessor zu takten, der nicht in der vorliegenden Anmeldung, sondern in den anderen anhängigen Anmeldungen dargestellt und erläutert ist. Die Taktung des Mikroprozessors erfolgt über die SCK-Leitung, die ebenfalls in den anhängigen Anmeldungen erläutert ist.

Dann folgt Block 616 und mit der Anstiegsflanke des SCK-Signals werden die Daten aus dem Benutzer-Mikroprozessor entnommen. Dann führt der Benutzer-Mikroprozessor die Daten auf den Bus 26, abhängig von einer möglichen Sperre durch die Arbitrations- oder Kollisionsdetektoren 42 und 44. Dies erfolgt im Block 618.

Mit der abfallenden Flanke eines SCK-Signals speichert der Benutzer-Mikroprozessor im Block 620 die Daten im Benutzer-Mikroprozessor, und es folgt Entscheidungsblock 622, in dem auf die 8. Abfallflanke im SCK-Signal gewartet wird. Gibt es keine 8. Abfallflanke, so kehrt das Verfahren zum Block 620 zurück, um wiederum Daten in den Mikroprozessor zu laden. Dies wird fortgesetzt, bis die 8. Abfallflanke des SCK-Signals im Entscheidungsblock 622 festgestellt wird, worauf Block 624 folgt, um die CONTROL-Leitung auf eine logische 1 zu setzen.

Dann wird die 1/2-Bitzeit im Block 626 geprüft und wenn diese erreicht ist, gibt die Schnittstelle IC 24 ein Stoppbit auf den Bus 26. Dies zeigt Block 628.

Im Block 630 prüft die Schnittstelle IC 24, ob die CONTROL-Leitung niedrig gesetzt worden ist. Wenn ja, so kehrt die Schnittstelle IC 24 zum Verfahren beginnend im Block 610 zurück, nachdem die CONTROL-Leitung im Block 636 niedrig gelatcht worden ist und wartet auf das Ende des Stoppbits im Block 638 und liefert dann ein Startbit an den Bus und setzt die IDLE-Leitung auf 1. Ist die CONTROL-Leitung nicht niedrig gesetzt worden, wie dies im Block 630 überprüft wird, so folgt Block 632, um zu prüfen, ob 11 freie Zeiten aufgetreten sind. Wenn nicht, so folgt die Rückkehr zum Block 630, um die CONTROL-Leitung erneut zu prüfen.

Sobald 11 freie Zeiten aufgetreten sind, folgt Block 634, um die IDLE-Leitung auf 0 zu setzen, damit die Übertragung im SPI-Modus zu beenden.

In Fig. 33 wird der SPI-Modus beim Datenempfang dargestellt.

Arbeitet die Schnittstelle IC 24 im SPI-Modus, so muß sie Daten empfangen. Die Schnittstelle IC 24 beginnt im Block 640 und dann folgt Block 642, um zu warten, bis ein Startbit am Bus 26 auftritt, bevor im Block 643 die IDLE-Leitung auf 1 gesetzt wird. Die Schnittstelle IC setzt dann die IDLE-Leitung auf eine logische 1.

Der Benutzer-Mikroprozessor wartet im Block 644 auf das Ende des Startbits und wenn dies der Fall ist, folgt Block 646, um die Schnittstelle IC für eine Übertragung zu sperren und die CONTROL-Leitung auf eine logische 0 zu setzen.

Sobald Abfallflanken des SCK-Signals auftreten, folgt Block 648, um vom Mikroprozessor die Daten in den Mikroprozessor zu latches. Die Busschnittstelle IC 24 im Block 650 wartet auf die 8. Abfallflanke des SCK-Signals und wiederholt das Latchen von Daten in den Benutzer-Mikroprozessor, s. Block 648, bis die 8. Abfallflanke des SCK-Signals erscheint. Zu diesem Zeitpunkt folgt Block 652, um die CONTROL-Leitung auf eine logische 1 zu setzen.

Nach der Zählung von $1/2$ Bitzeiten im Entscheidungsblock 654 folgt Block 656, um eine Stoppbit-Periode zu empfangen. Sobald das Ende des Stoppbits im Block 658 festgestellt worden ist, folgt Block 660, um zu prüfen, ob es ein anderes Startbit auf dem Bus 26 gibt. Ist dies der Fall, so kehrt das Verfahren zum Block 644 zurück und wird wiederholt, wie vorbeschrieben. Falls nicht, so folgt Block 662, in dem auf das Auftreten von zehn freien Zeiten gewartet wird. Trifft dies ein und liegt kein anderes Startbit am Bus an, so folgt Block 664, um die IDLE-Leitung auf 0 zu setzen und damit die Sperre für den Zugang der Schnittstelle IC an den Bus 26 zu unterbrechen.

Damit ist der SPI-Modus beim Datenempfang beendet.

In den Fig. 34 und 35 wird der gepufferte SPI-Modus bei der Übertragung und im Empfang erläutert.

Wird für eine Datenübertragung der gepufferte SPI-Modus in Block 700 eingegeben, so wird im Block 702 die CONTROL-Leitung überprüft, wann sie hochpegelig wird.

Sobald die CONTROL-Leitung hochpegelig wird, gibt der Benutzer-Mikroprozessor (der wie bereits erwähnt, in anderen anhängigen Anmeldungen dargestellt ist) 16 Bits in die Schnittstelle IC 24 im Block 704.

Dann wird im Block 706 die CONTROL-Leitung überprüft, um festzustellen, wann sie niederpegelig wird. Sobald dies der Fall ist, folgt Block 708, um den CONTROL-Pin niederpegelig zu latchen und dann im Block 710 zu prüfen, ob das Startbit auf dem Bus 26 anliegt. Ist dies nicht der Fall, so folgt Block 712, in dem die IDLE-Leitung geprüft wird, ob diese mindestens zwei Bitzeiten lang niedrig ist. Ist sie dies nicht, so kehrt das Verfahren zum Block 710 zurück, um erneut zu prüfen, ob am Bus 26 das Startbit anliegt.

Ist aber die IDLE-Leitung während mindestens zwei Bitzeiten niedrig, so folgt Block 714, um ein Startbit zu erzeugen und dann den IDLE-Zustand auf eine logische 1 im Block 716 zu setzen.

Liegt aber auf dem Bus 26 ein Startbit, so folgt auf Block 710 unmittelbar der Block 716, um IDLE auf eine logische 1 zu setzen. Dann folgt Block 718, um das Ende des Startbits festzustellen und zu warten, bis dieses eintritt, worauf Block 720 folgt, um 16 Datenbits zum Bus 26 getrennt von Start- und Stoppbits zu führen. Gleichzeitig im Block 722 die Daten vom Arbitrationsdetektor 42 und Kollisionsdetektor 44 gesperrt werden. Der Benutzer-Mikroprozessor 22 wird dann vom Puffer 304 der Schnittstelle IC 24 gemäß Block 724 abgetrennt.

Sobald der Block 726 feststellt, daß das zweite Stoppbit aufgetreten ist, so wird im Block 728 die CONTROL-Leitung auf eine logische 1 gesetzt, bevor die Daten im Puffer 304 arrangiert werden, um gemäß Block 730 das erste Byte mit dem höchstwertigen Bit voraus zuerst zum Mikroprozessor zu übertragen.

Die Schnittstelle IC 24 wird dann im Block 732 vom Puffer 304 abgetrennt.

Der Entscheidungsblock 734 wartet auf das Lesen von 16 Bits. Sobald 16 Bits gelesen worden sind, prüft die Schnittstelle IC im Block 736, ob die CONTROL-Leitung auf 0 Pegel ist. Ist dies der Fall, so kehrt die Schnittstelle IC zum Block 708 zurück, um den CONTROL-Pin niedrig zu latchen und dann folgt das dem Block 708 folgende Verfahren.

Ist aber die CONTROL-Leitung nicht auf einer logischen 0, was der Block 736 feststellt, so folgt Block 738, um zu prüfen, ob 11 Bitzeiten aufgetreten sind. Wenn nicht, so kehrt die Schnittstelle IC 24 zum Block 734 zurück und prüft erneut, ob 16 Bits gelesen worden sind. Sind die 11 Bitzeiten erreicht, so ändert die Übertragung im gepufferten SPI-Modus, nachdem im Block 740 die IDLE-Leitung auf eine logische 0 gesetzt wird.

Wenn aber im Block 734 16 Bits nicht gelesen worden sind, so folgt unmittelbar Block 738, um das Auftreten von 11 Bitzeiten zu prüfen und dann folgt der Rest des Verfahrens von dort.

In Fig. 35 ist der gepufferte SPI-Modus im Empfangszustand dargestellt, der im Block 760 beginnt, worauf der Entscheidungsblock 762 folgt, indem festgestellt wird, ob das Startbit am Bus 26 ansteht. Sobald dies eintritt, folgt Block 764, um IDLE auf eine logische 1 zu setzen, bevor auf das Ende des Startbits im Block 766 gewartet wird. Die Schnittstelle IC wartet auf das Ende des Startbits, worauf dann das weitere Verfahren folgt.

Für die Schnittstelle IC 24 folgt dann der Entscheidungsblock 768, um zu prüfen, ob 16 Bits gelesen worden sind. Wenn nicht, so wird der Puffer 304 am Zugang zum Bus 26 im Block 770 gesperrt und die Schnittstelle IC wartet auf einen freien Zustand des Bus, indem die IDLE-Leitung im Block 772 geprüft und gewartet wird, bis sie niederpegelig

wird, bevor sie zu dem bereits erläuterten Entscheidungsblock zurückkehrt und die Schritte ab dem Block 762 ausführt, um den Datenbus erneut zu prüfen, ob ein Startbit ansteht.

Wenn dagegen im Entscheidungsblock 768 16 Bits gelesen worden sind, so folgt Block 774, um die CONTROL-Leitung niederpegelig zu setzen und den Puffer 304 zu takten. Im Block 776 gehen acht Datenbits und Stoppbit auf den Bus 26. Nach dem Ende des Stoppbits prüft die Schnittstelle IC im Block 778 erneut, ob ein Startbit am Datenbus 26 ansteht. Wenn nicht, so prüft die Schnittstelle im Block 780, ob 10 freie Bitzeiten aufgetreten sind und kehrt dann zurück, um das Auftreten eines Startbits am Bus 26 im Block 778 erneut zu prüfen, falls 10 freie Bitzeiten nicht aufgetreten sind.

Wenn 10 freie Bitzeiten gezählt worden sind, so wird eine Byte-Flag gesetzt, bevor Daten aus der Mitte des Puffers 304 im Block 782 entnommen werden. Dann wird die IDLE-Leitung im Block 784 niederpegelig gesetzt, nachdem der Puffer 304 vom Benutzer-Mikroprozessor 22 im Block 783 getaktet wird.

Wenn jedoch im Block 778 festgestellt wird, daß ein Startbit am Bus 26 erscheint, dann dürfen im Block 786 acht weitere Bits auf den Bus 26 gehen. Im Block 788 prüft die Schnittstelle IC 24 die Stoppbitzeit und wartet, bis diese erfolgt, worauf Block 790 folgt, um die CONTROL-Leitung auf eine logische 1 zu setzen, bevor der Puffer 304 vom Benutzer-Mikroprozessor 22 im Block 791 getaktet wird.

Im Block 792 wird auf das Ende des Stoppbits gewartet und sobald dies der Fall ist, folgt Block 794, um zu prüfen, ob ein Startbit am Bus 26 anliegt. Wenn ja, so kehrt die Schnittstelle IC zu dem bereits beschriebenen Verfahren, beginnend im Block 766 zurück.

Wenn im Block 794 festgestellt wird, daß kein Startbit am Bus 26 ansteht, so folgt Block 796, um das Vorhandensein von 10 freien Bitzeiten zu prüfen, und um zum Block 794 zurückzukehren, wenn keine 10 freien Bits vorliegen.

Sobald jedoch 10 freie Bitzeiten im Block 796 getaktet worden sind, so folgt Block 798, um die IDLE-Leitung niederpegelig zu setzen.

Man bemerke, daß der Puffer 304 voll ist, wenn die CONTROL-Leitung hochpegelig ist, während der Puffer 304 geladen wird, wenn sie niederpegelig ist.

EP 87 107 565.1-2201
CHRYSLER CORPORATION

M-6974

PATENTANSPRÜCHE:

1. Kommunikationssystem zur Übertragung von Nachrichten in Datenform über einen Datenbus (26) zwischen zwei oder mehreren Benutzer-Mikroprozessoren (22), die an den Datenbus über eine IC-Busschnittstelle (24) angeschlossen sind, wobei die IC-Busschnittstelle aufweist:
einen Arbitrationsdetektor (42);
einen Kollisionsdetektor (44);
einen Bustreiber (28);
einen Busempfänger (30);
eine Stromquelle (34) und eine Stromsenke (36), um den logischen Zustand des Bus zu schalten, abhängig von Signalen vom Bustreiber und Busempfänger der zu jedem Benutzer-Mikroprozessor am Bus zugehörigen IC-Busschnittstelle, und einen Leerlauf- und Steuerdetektor (54), wobei der Busempfänger (30) zwischen dem Datenbus (26) und dem Schnittstellenanschluß des Benutzer-Mikroprozessors (22) geschaltet ist, um Nachrichten in Datenform vom Datenbus zum Benutzer-Mikroprozessor zu empfangen;
wobei der Leerlauf- und Steuerdetektor (54) zwischen dem Eingangs-/Ausgangsanschluß des Benutzer-Mikroprozessors (22) und individuell an den Arbitrationsdetektor (42) und den Kollisionsdetektor (44) angeschlossen ist, um den Datenbus (26) zu überwachen und festzustellen, wann der Datenbus frei und wann der Datenbus belegt ist;
wobei das Kommunikationssystem folgendermaßen gekennzeichnet ist:
die IC-Busschnittstelle (24) ist fähig, Daten von einem Benutzer-Mikroprozessor (22) formiert für Datenverbindungen im Modus SCI, SPI oder BSPI zu empfangen, wobei die IC-Busschnittstelle (24) ferner aufweist:

eine Modusauswahlschaltung (301) zum Steuern des Daten- und Schiebetakt (SCK)-Signalflusses in und aus der IC-Bus-schnittstelle in Antwort auf die spezielle Schnittstelle des Benutzer-Mikroprozessors und mit einem Betrieb gemäß dieser Schnittstelle SCI, SPI oder BSPI;

einen Puffer (304);

einen SCK-Selektor (302) zum Dirigieren von Daten in einem gepufferten SPI-Modus und zum Beschreiben der Quelle des Taktsignals für den Puffer (304);

einen SCK-Zähler (303) zum Zählen der SCK-Impulse vom Benutzer-Mikroprozessor;

einen Byte-Zähler (306) zum Zählen der empfangenen Anzahl von Bytes;

einen Start/Stop-Schiebetaktsignal-(SCK) Generator (307);

einen SPI-Taktsignal-Generator (308) zum Erzeugen von SCK-Impulsen für den Benutzer-Mikroprozessor und Puffer (304);

ein SPI-Übertrager-Planungs-(309) und Steuergerät, um zu steuern, wann Daten von einem Benutzer-Mikroprozessor in den Bus im SPI- oder BSPI-Modus gespeist werden;

wobei die aus der Verbindung des Start/Stop-SCK-Generators (307), des Arbitrationsdetektors (42), des Kollisionsdetektors (44) und des Bustreibers (28) gebildete Reihenschaltung zwischen den Schnittstellenanschluß des Benutzer-Mikroprozessors (22) und den Datenbus (26) geschaltet ist, um Daten vom Benutzer-Mikroprozessor auf den Datenbus zu übertragen;

wobei die Benutzer-Mikroprozessoren (22) einen Schnittstellenanschluß der folgenden Typen besitzen: einen (SCI)-Anschluß, einen (SP)I-Anschluß und einen gepufferten, seriellen peripheren Schnittstellen-(BSP)I-Anschluß mit einem Taktanschluß und einem Eingangs-/Ausgangsanschluß, und wobei der Puffer (304) zwischen den Schnittstellenanschluß des Benutzer-Mikroprozessors (22) und der Serienschaltung geschaltet ist.

2. Kommunikationssystem nach Anspruch 1, das ferner aufweist:

ein zwischen den Busempfänger(30) und den SCI-Anschluß des Benutzer-Mikroprozessors (22) geschaltetes digitales Filter (210), um Rauschen aus den Nachrichten in Datenform auszufiltern, bevor diese in anderen Teilen der IC-Busschnittstellen IC verarbeitet werden;

Zeit- und Synchronisiereinrichtungen zum Synchronisieren und zum Erzeugen eines Baud-Zeitsignals für den Arbitrationsdetektor, den Kollisionsdetektor und den Leerlauf-Detektor, bestehend aus

einem Taktsignalteil (201) zum Empfangen eines Taktimpulses vom Benutzer-Mikroprozessor und zum Erzeugen eines Taktsignals;

einem Wortzähler (202) zum Annehmen des Taktsignals und zum Anbieten von Zeit- und Synchronisierungsinformation an den Arbitrationsdetektor (42) und den Kollisionsdetektor (44);

einem Start-Bit-Detektor (202) zur Anzeige, wann ein Start-Bit-Signal und Pegel auf den Datenbus vorhanden ist;

einem Rahmenfehlerdetektor (204) zur Anzeige, wenn der Stop-Bit-Pegel auf dem Datenbus nicht zur Stop-Bit-Zeit vorhanden ist.

3. Kommunikationssystem nach Anspruch 1 oder 2, das ferner aufweist:

Einen Unterbrecher-Generator (403), der es einem Benutzer-Mikroprozessor erlaubt, auf dem Bus eine logische Bedingung zu erzwingen.

4. Kommunikationssystem nach einem der Ansprüche 1 bis 3, das ferner aufweist:

Eine Überbereichssperre (61) und einen Überbereichsdetektor (60) zum Feststellen, ob die Spannung am Bus von einem vorbestimmten Spannungsfenster abweicht und zur Anzeige einer auftretenden Abweichung.

5. Kommunikationssystem nach Anspruch 4, das ferner aufweist:

Einen Testmodusdetektor (401), der der Modusauswahlschaltung (301) signalisiert, daß sie Daten aus der Überbereichssperre (61) zu dem Benutzer-Mikroprozessor (22) überträgt, und der dem Benutzer-Mikroprozessor gestattet, eine Rückstellung auszulösen.

6. Für ein Kommunikationssystem nach einem der Ansprüche 1 bis 5, Verfahren zum Übertragen von Daten in einem SPI-Modus des Betriebes in Verbindung mit einem Verfahren zum Arbitrieren von Daten auf dem Datenbus, das aufweist:

Überprüfen, ob der Datenbus (26) frei ist oder Warten, falls nicht:

Abwarten, daß der Bus für eine 2-Bit-Zeit frei ist;

Aufschalten eines Start-Bits auf den Datenbus;

Warten bis zum Ende des Start-Bits;

Takten des Benutzer-Mikroprozessors (22);

Abrufen von Daten aus dem Benutzer-Mikroprozessor;

Aufschalten der Daten vom Benutzer-Mikroprozessor auf den Bus, abhängig von einer möglichen Sperre von Seiten der von der IC-Busschnittstelle benutzten Arbitrations- oder Kollisionsverfahren;

Einspeichern der Daten in den Benutzer-Mikroprozessor;

Warten auf die achte abfallende Kante eines Taktsignals;

Warten auf das Auftreten einer 1/2-Bit-Zeit;

Aufschalten eines Stop-Bits auf den Datenbus;

Überprüfen, ob der Benutzer-Mikroprozessor ein anderes Byte übertragen möchte;

wenn der Benutzer-Mikroprozessor ein anderes Byte nicht zu übertragen wünscht, Warten auf das Auftreten von elf Freizeiten und Durchführen einer weiteren Überprüfung, ob der Benutzer-Mikroprozessor mehr Daten auf den Datenbus zu übertragen wünscht; und

wenn der Benutzer-Mikroprozessor ein weiteres Byte zu über-

tragen wünscht, Rückkehr zu den vorstehenden Schritten für die Übertragung der Daten.

7. Für ein Kommunikationssystem nach einem der Ansprüche 1 bis 5, Verfahren zum Empfangen von Daten in einem SPI-Modus des Betriebes in Verbindung mit einem Verfahren zum Arbitrieren von Daten auf den Datenbus, das aufweist:

- Überwachen, ob auf dem Datenbus ein Start-Bit erscheint;
- Warten bis zum Ende des Start-Bits, wenn ein Start-Bit auf dem Bus erscheint;
- Signalisieren des Arbitrationsverfahrens, um das Übertragen über die Bus-Schnittstellen-IC zu sperren;
- Einspeichern von Daten in den Benutzer-Mikroprozessor;
- Warten auf die achte abfallende Kante des Taktsignals und dann Zählen von 2-Bit-Zeiten;
- Empfangen eines Stop-Bits und Warten auf das Ende des Stop-Bits, das erscheint;
- Prüfen, ob ein weiteres Start-Bit auf dem Datenbus erscheint;
- Wiederholen der vorstehenden Schritte, um mehr Daten einzuspeichern, wenn ein weiteres Start-Bit auf dem Datenbus ansteht;
- Abwarten von zehn Frei-Zeiten, wenn kein weiteres Start-Bit auf dem Datenbus ansteht und Ansprechen der Arbitrationseinrichtungen, um die IC Busschnittstelle aus der Datenübertragung freizugeben.

8. Für ein Kommunikationssystem nach einem der Ansprüche 1 bis 5, Verfahren zum Übertragen von Daten in einem gepufferten SPI-Modus des Betriebes in Verbindung mit einem Verfahren zum Arbitrieren von Daten auf den Datenbus, das aufweist:

- Prüfen, ob ein weiterer Benutzer-Mikroprozessor höherer Priorität bereits eine Übertragung auf dem Datenbus durchführt und Warten, wenn dies der Fall ist;

wenn der Datenbus zur Übertragung verfügbar ist, Empfangen
 des Dateneingangs aus dem Benutzer-Mikroprozessor;
 Ansprechen des Datenbus, daß die Übertragung jetzt statt-
 finden soll;
 Prüfen, ob ein Start-Bit auf dem Datenbus ansteht und Ab-
 warten von mindestens zwei Frei-Bit-Zeiten, wenn kein
 Start-Bit auf dem Datenbus ansteht, um ein Stop-Bit zu er-
 zeugen;
 Senden eines Signals auf den Datenbus zur Anzeige, daß der
 Bus belegt ist;
 Warten auf das Enden des Start-Bits;
 Aussenden von durch Start- und Stop-Bits getrennte Daten
 auf den Bus, wenn dies nicht von Arbitrations-/Kollisions-
 verfahren in der IC-Busschnittstelle gesperrt ist;
 Trennen des Benutzer-Mikroprozessors vom Puffer an der IC-
 Busschnittstelle;
 Warten auf das Auftreten des zweiten Stop-Bits;
 Anordnen von Daten im Puffer zum Einspeichern des ersten
 Bytes derart, daß das Byte mit der größten Signifikanz aus
 dem Benutzer-Mikroprozessor zuerst gesendet wird;
 Trennen der IC-Busschnittstelle vom Puffer;
 Warten auf das Auslesen der Daten;
 Wiederholen der vorstehenden Schritte, wenn mehr Daten
 übertragen werden sollen;
 Abwarten von elf Bit-Zeiten, wenn die Datenübertragung
 vollständig ist und Freigabe des Bus in den freien Zustand.

9. Für ein Kommunikationssystem nach einem der
 Ansprüche 1 bis 5, Verfahren zum Empfangen von Daten in
 einem gepufferten SPI-Modus des Betriebes in Verbindung mit
 einem Verfahren zum Arbitrieren von Daten auf den Datenbus,
 das aufweist:

Warten auf das Erscheinen eines Start-Bits auf den Daten-
 bus;

Ansprechen anderer Benutzer-Mikroprozessoren, daß der Datenbus belegt ist;
Warten auf das Auftreten des Endes des Start-Bits;
Warten auf zu lesende Daten;
Trennen des Puffers vom Bus, wenn die Daten nicht gelesen worden sind und Warten, daß der Bus in den freien Zustand zurückkehrt, bevor nochmals geprüft wird, ob ein Start-Bit auf dem Datenbus aufgetreten ist;
wenn alle Daten in den Puffer eingelesen worden sind, Takten des Puffers;
Freigeben, daß Daten und Stop-Bit auf den Datenbus gelangen;
Freigeben, daß der Benutzer-Mikroprozessor den Puffer taktet;
Warten auf das Ende des Stop-Bits;
Warten auf das Auftreten des Start-Bits auf den Datenbus vor dem Auslesen der restlichen Daten vom Bus, aber Freigabe des Datenbus in den freien Zustand, wenn zehn Frei-Zeiten dem ersten Eingang von Daten folgen;
Warten auf das Auftreten der Stop-Bit-Zeit und Warten auf das Ende des Stop-Bits;
Prüfen, ob ein Start-Bit auf dem Bus aufgetreten ist und Wiederholen der obigen Vorgänge, wenn eine weitere Übertragung auftritt;
Warten auf zehn Frei-Bit-Zeiten, wenn ein Start-Bit auf dem Bus nicht ansteht und Mitteilen an die restlichen Benutzer-Mikroprozessoren, daß der Bus in einem freien Zustand ist.

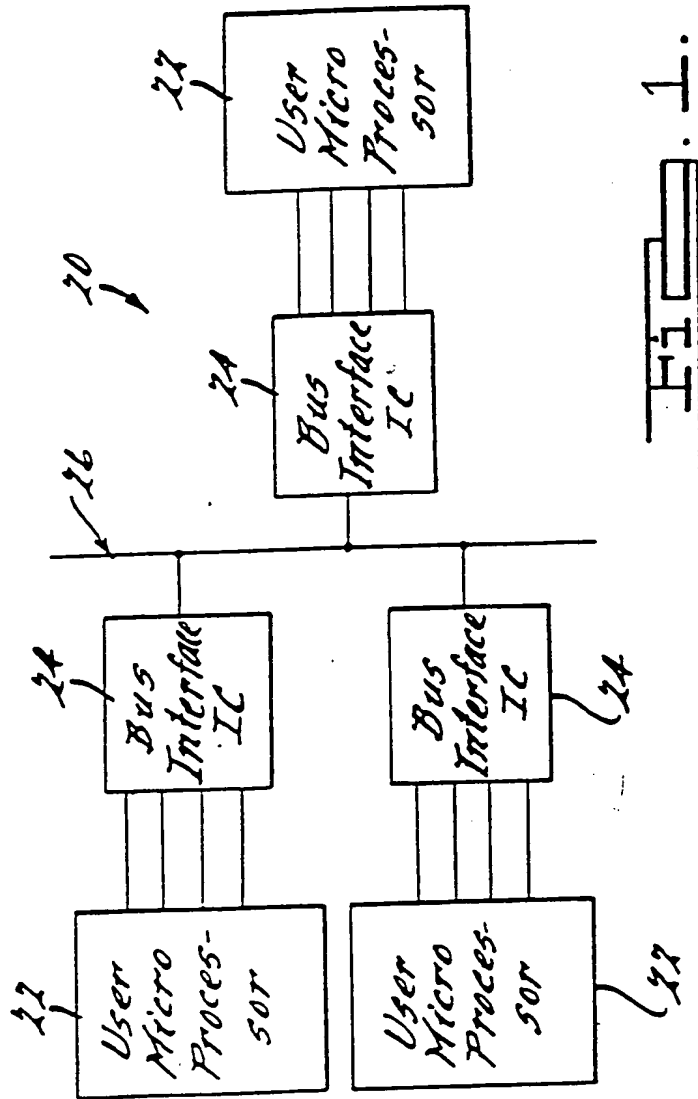
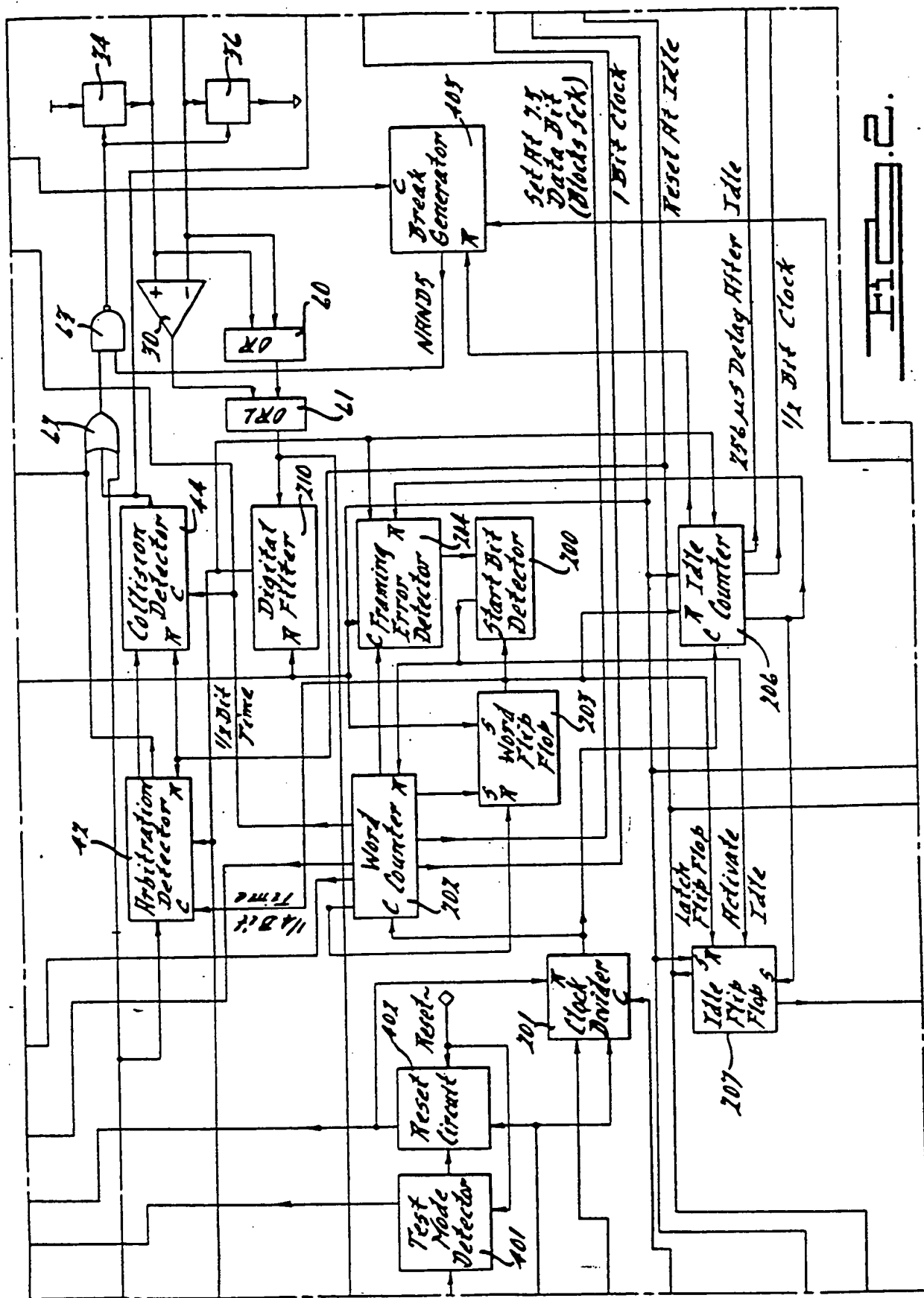
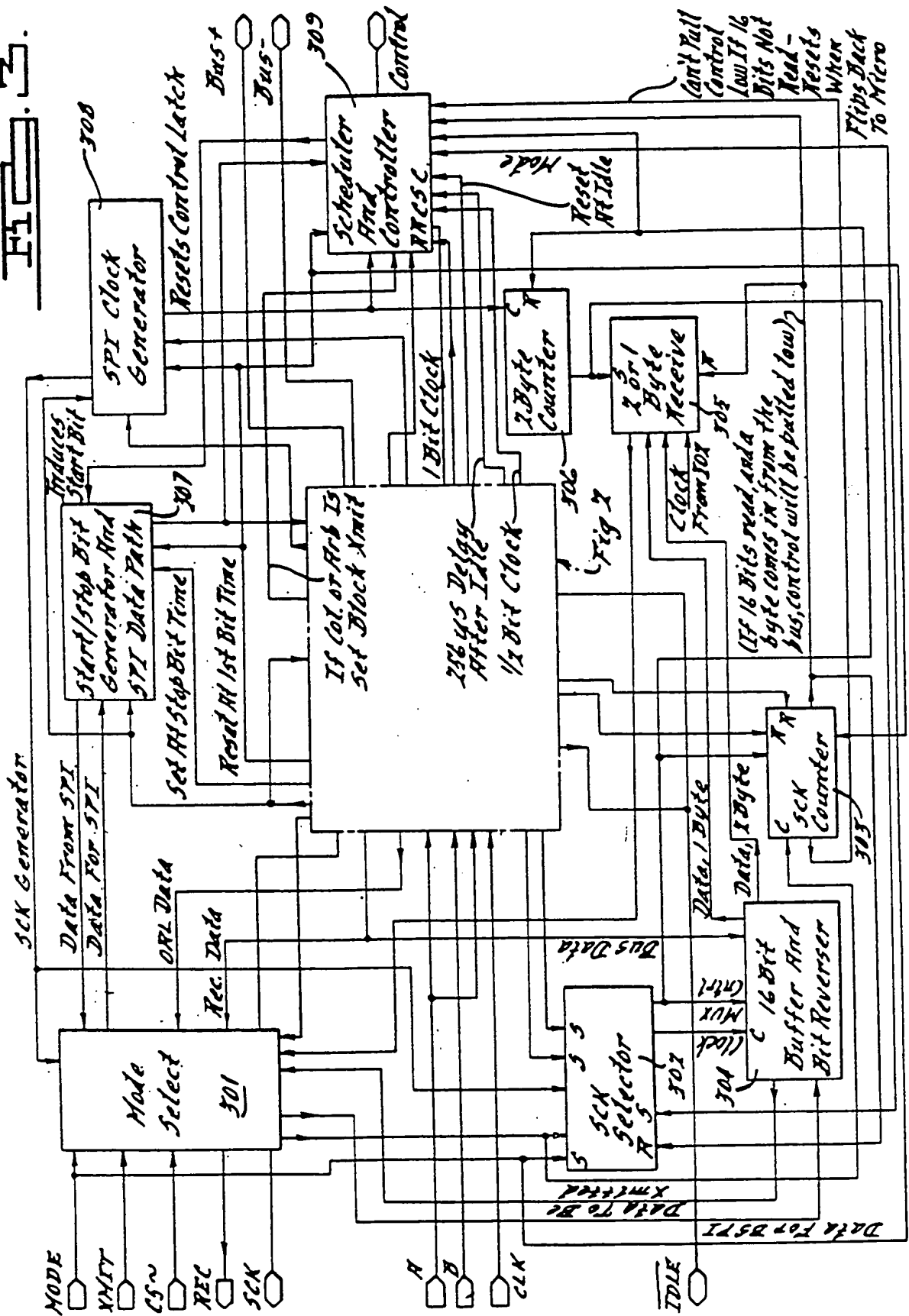


FIG. 1.



五五五



4/22

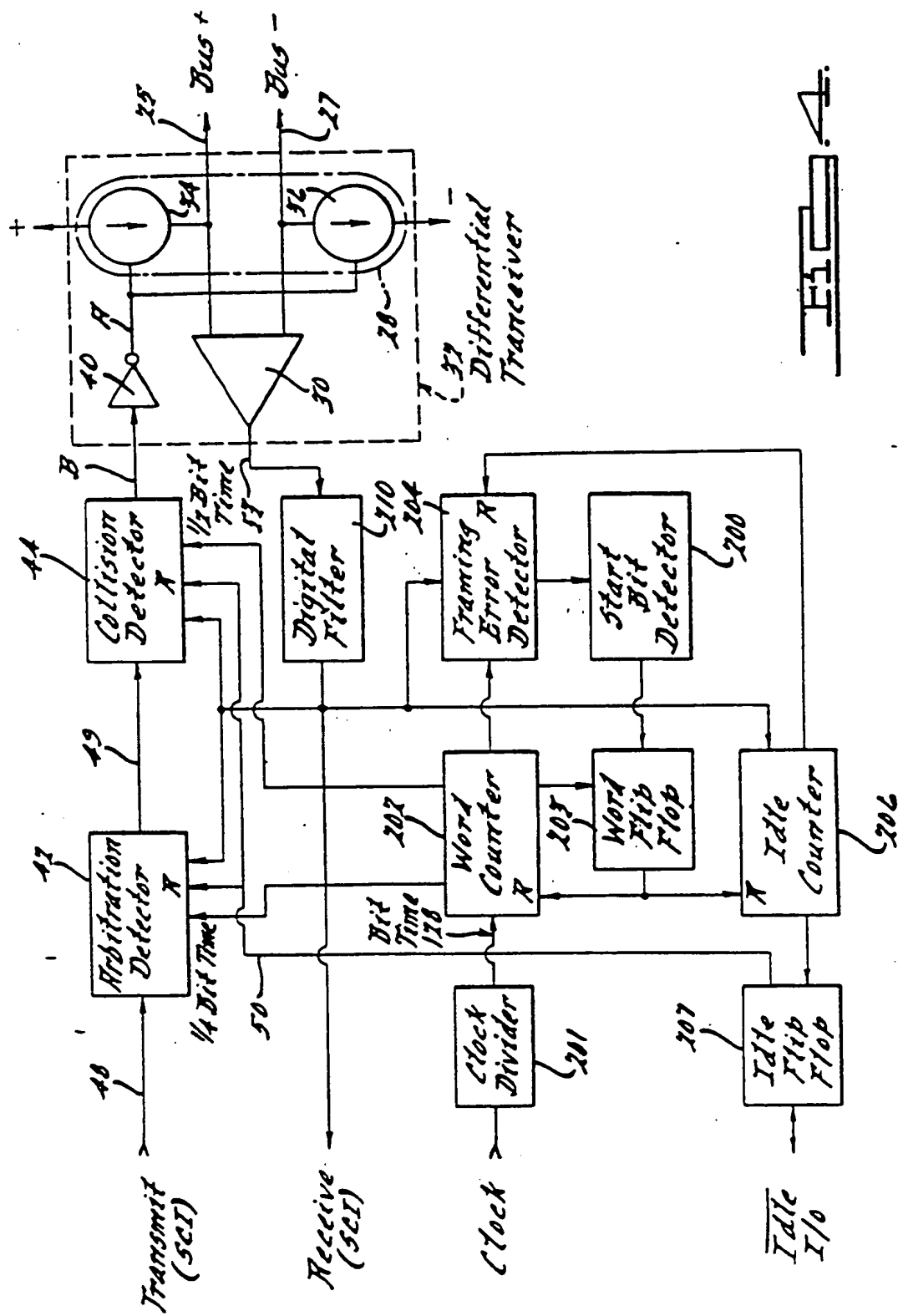
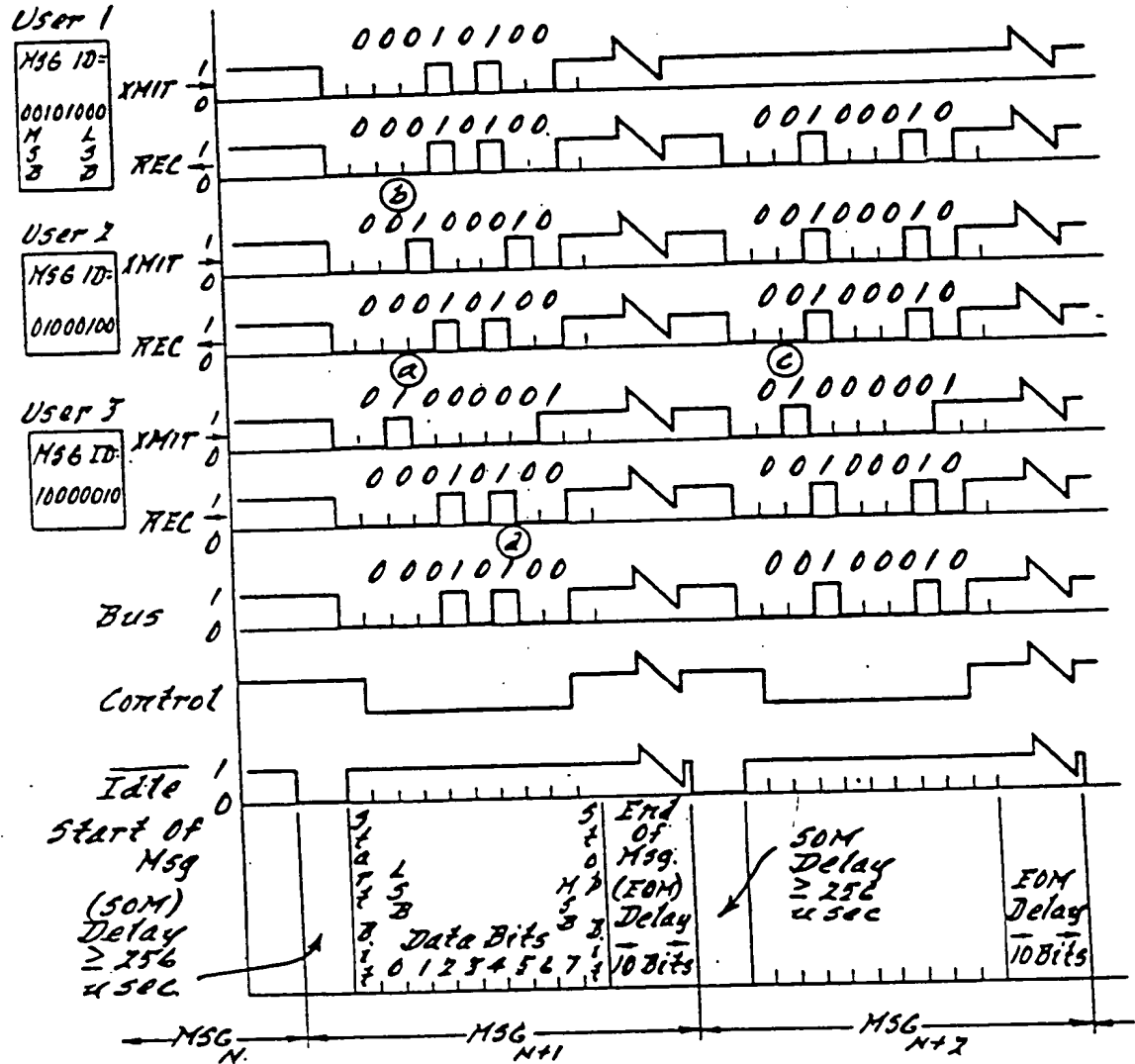


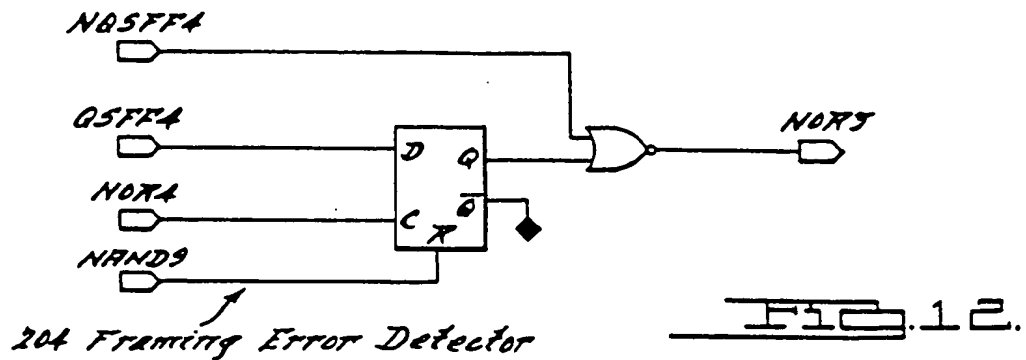
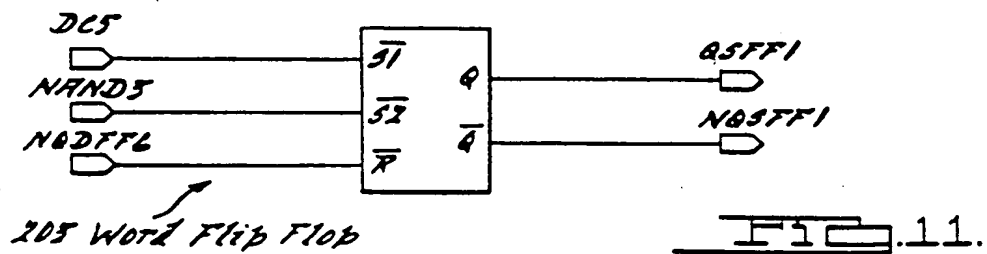
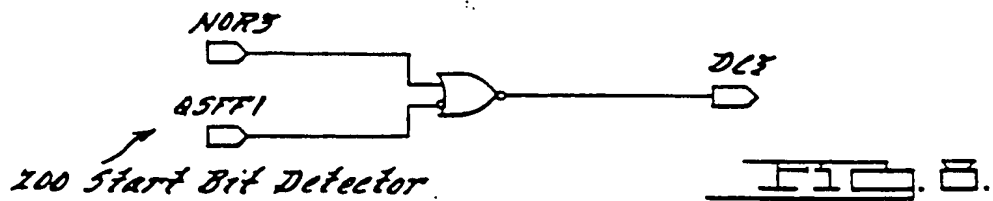
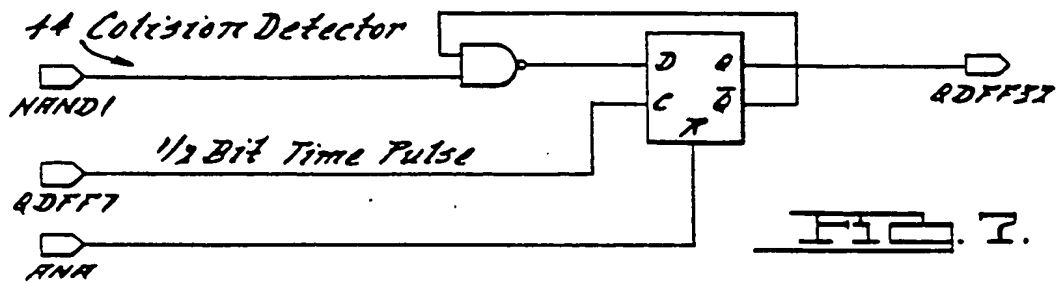
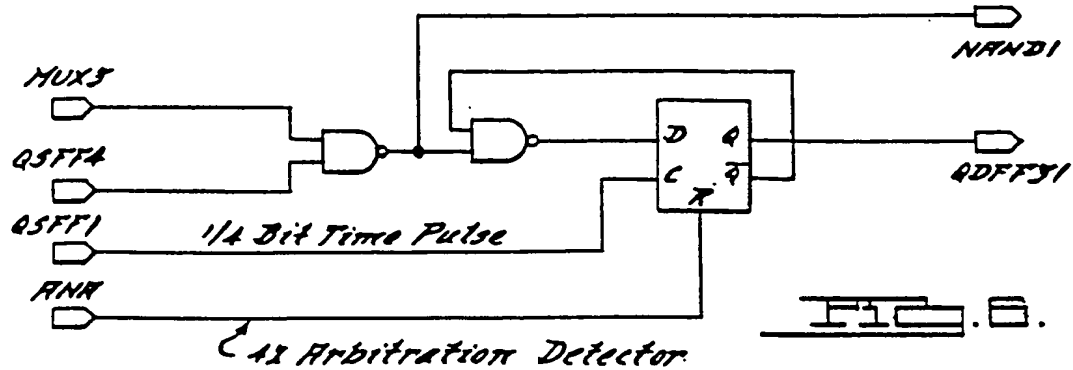
FIG. 4

5/22

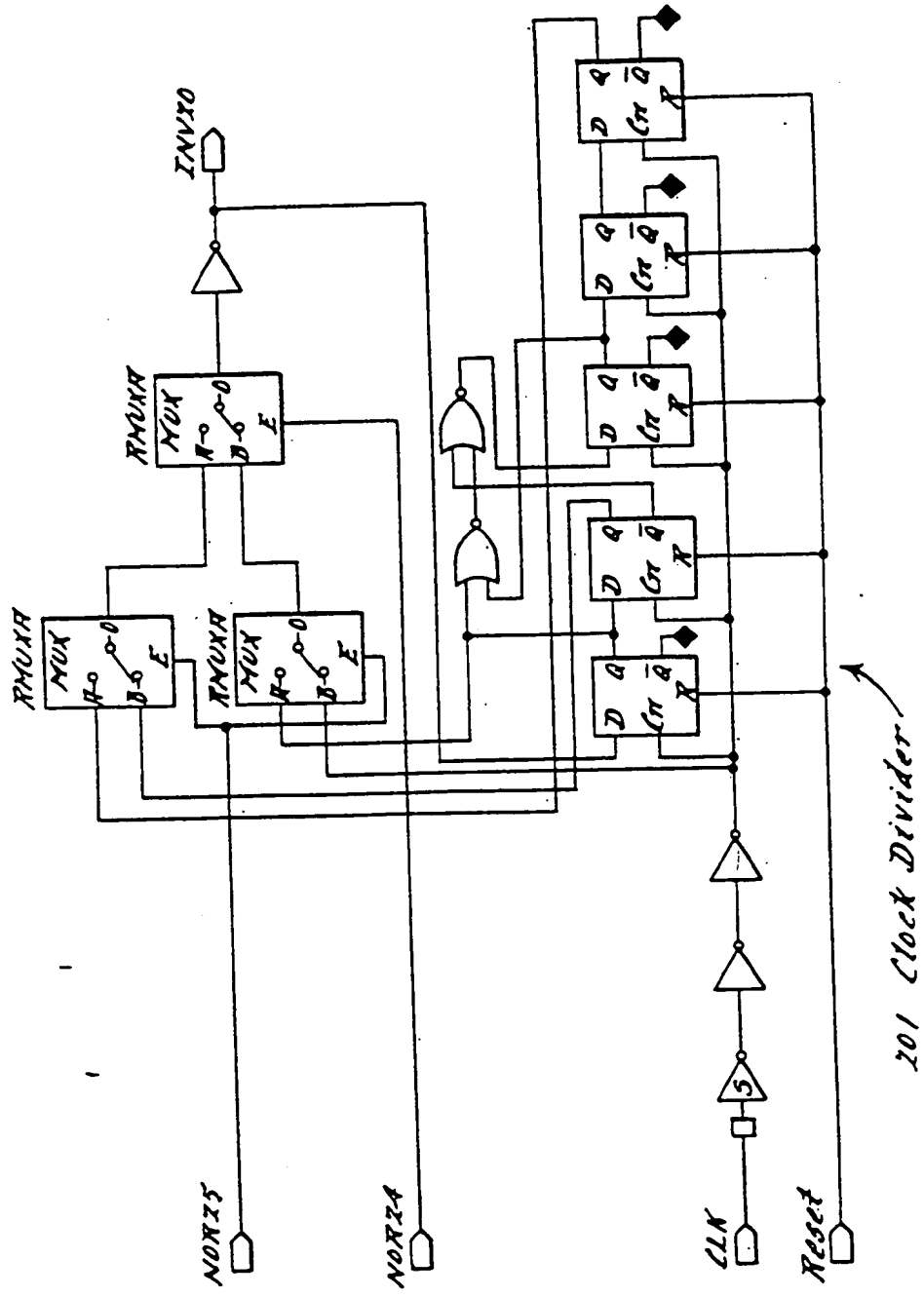
FIG. 5.



6/22

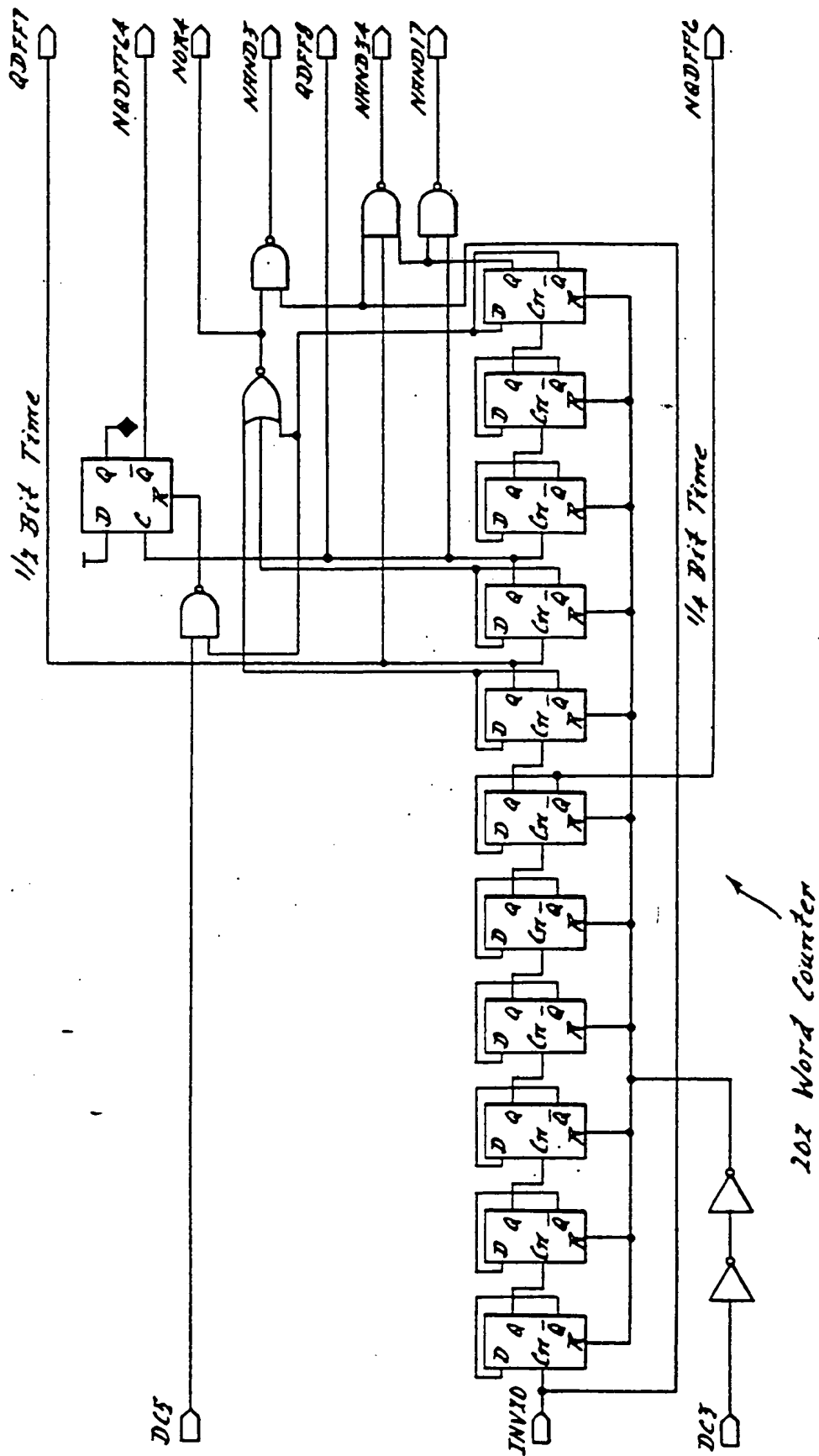


22/9

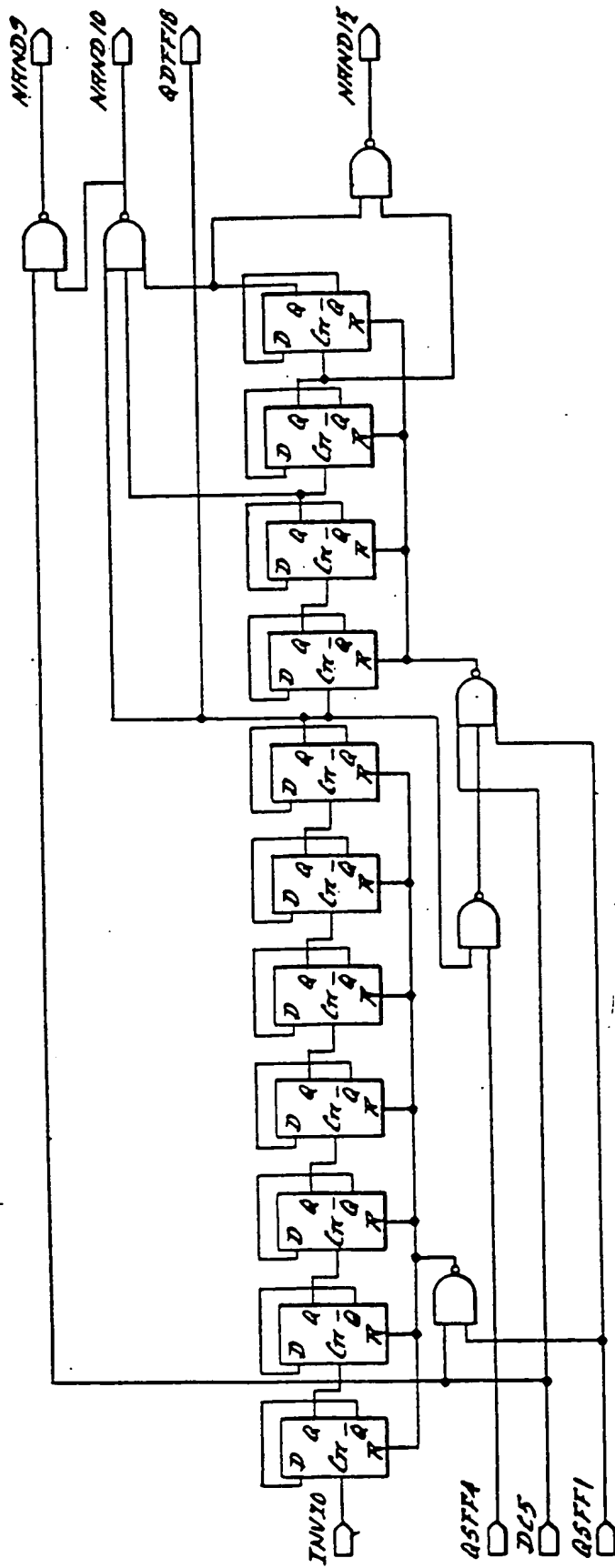


201 Clock Divider

10.10.



9/22



206 Idle Counter

FIG. 13.

10/22

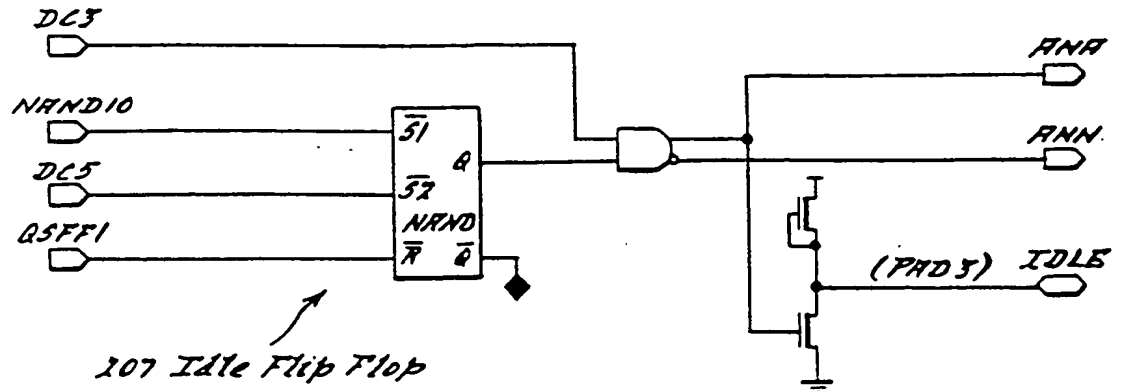


FIG. 14.

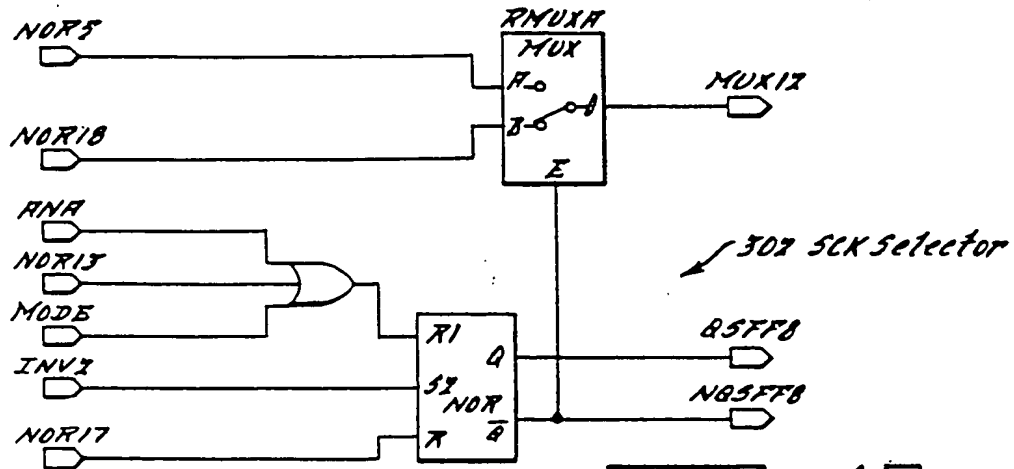


FIG. 17.

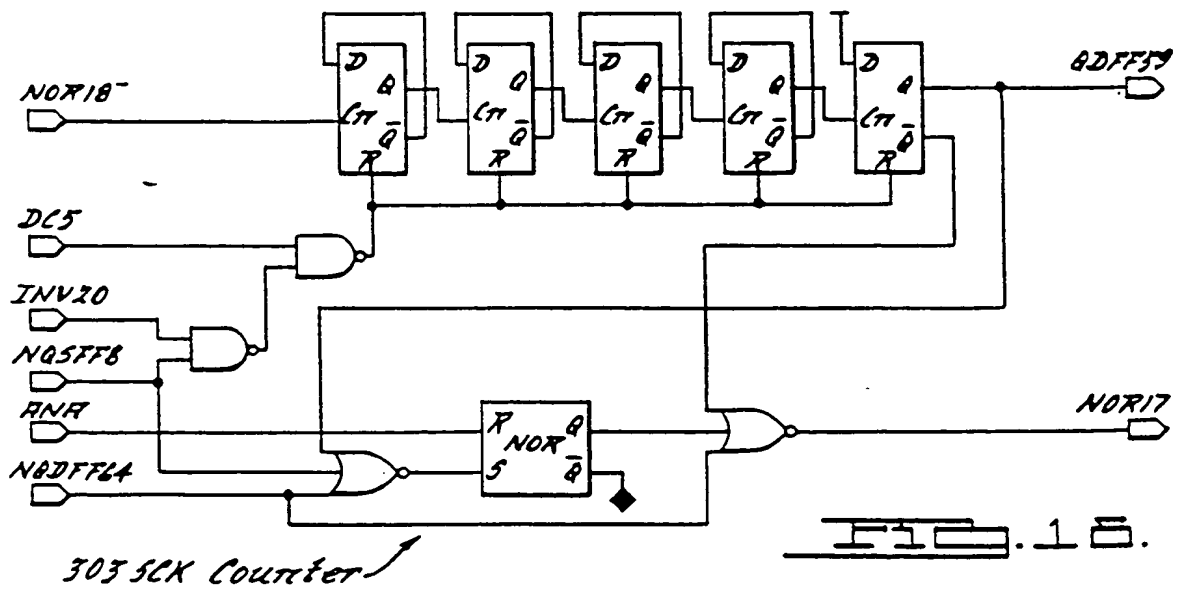


FIG. 18.

22/11

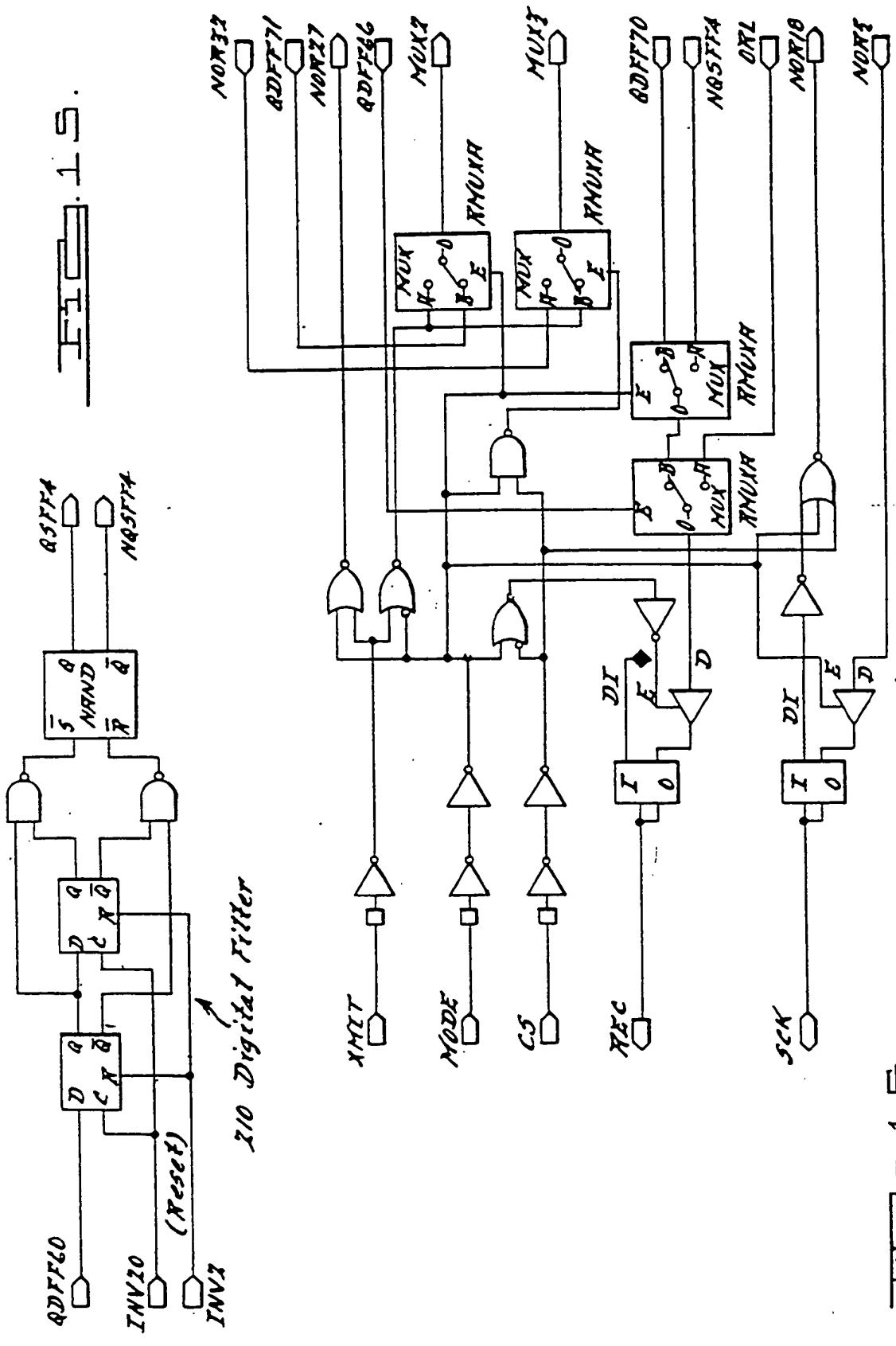
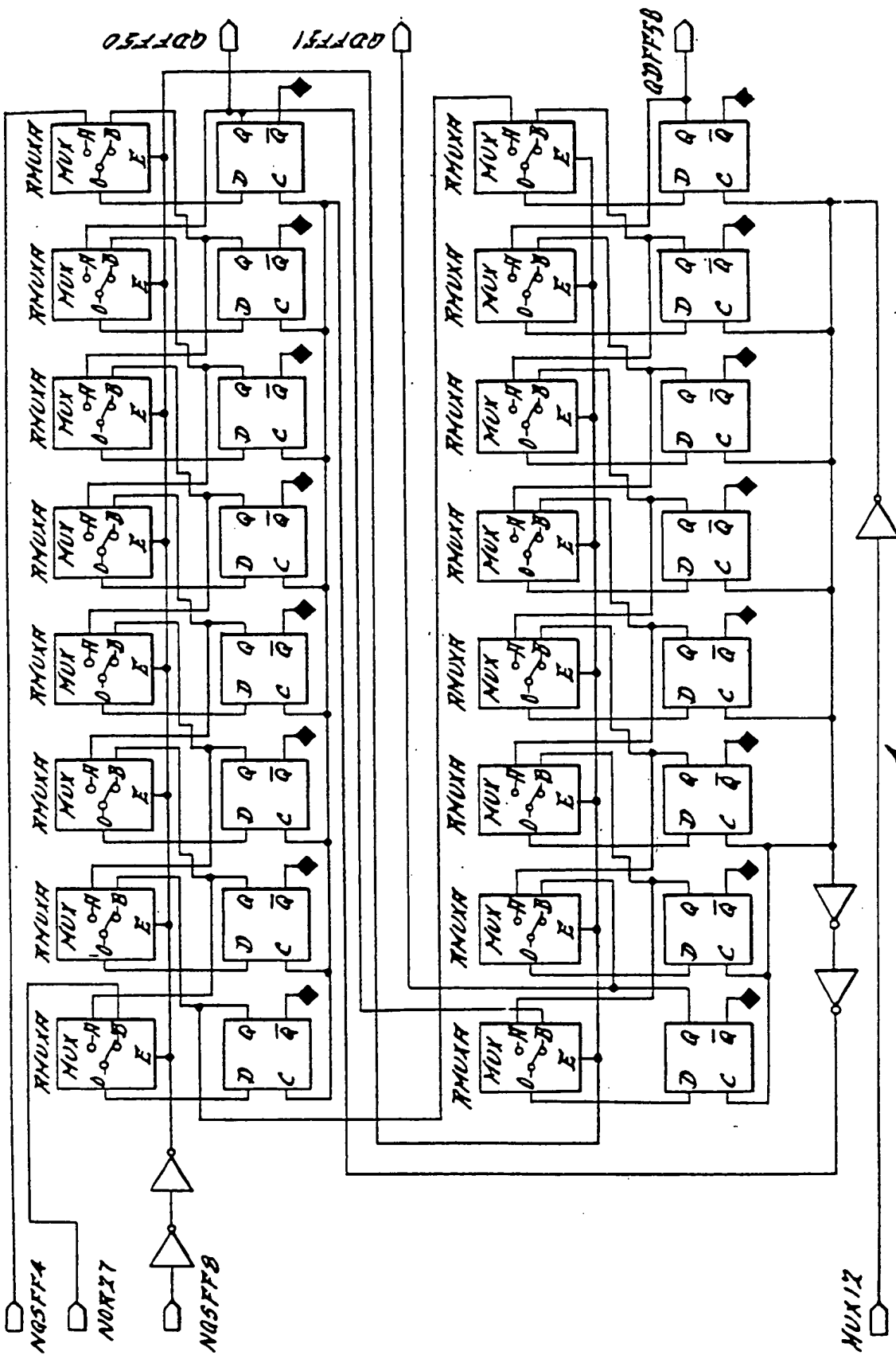


FIG. 15.

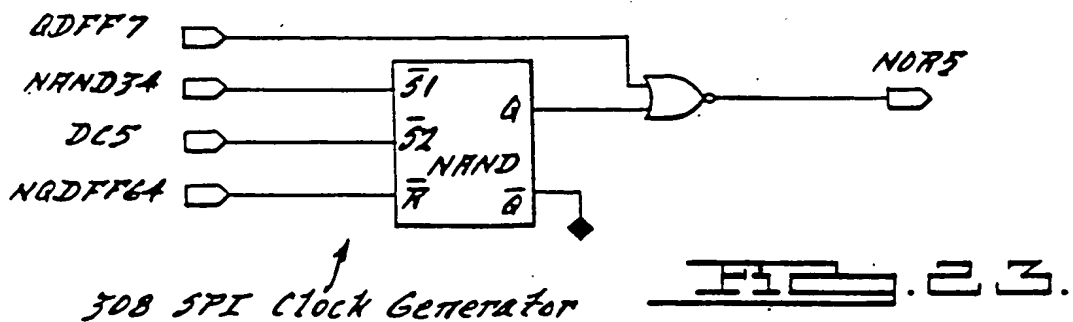
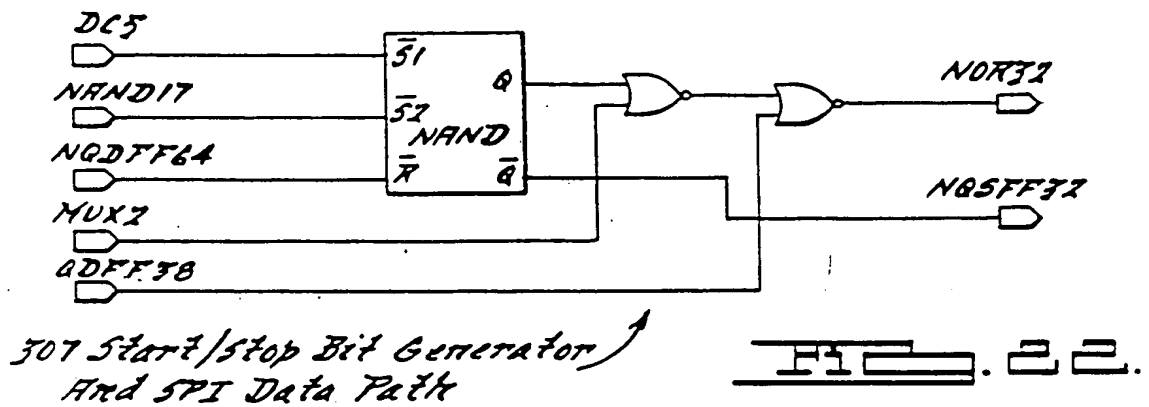
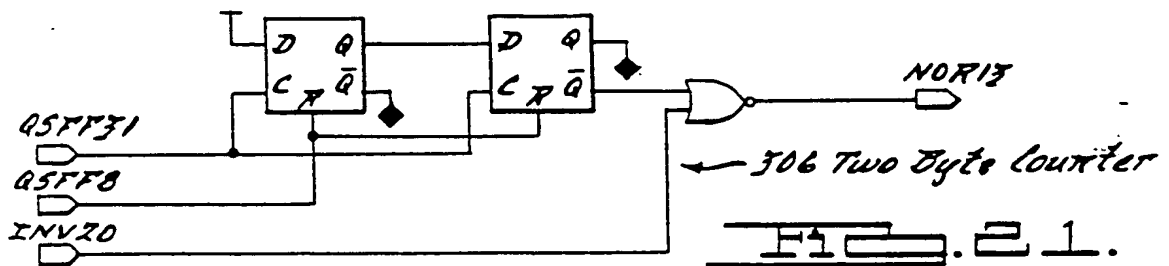
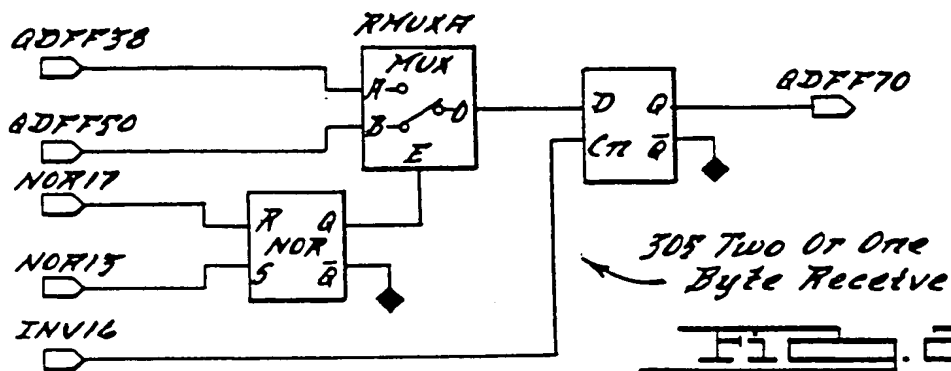
FIG. 16.

22/12



304 Sixteen Bit Buffer And Bit Reverser

13/22



14/22

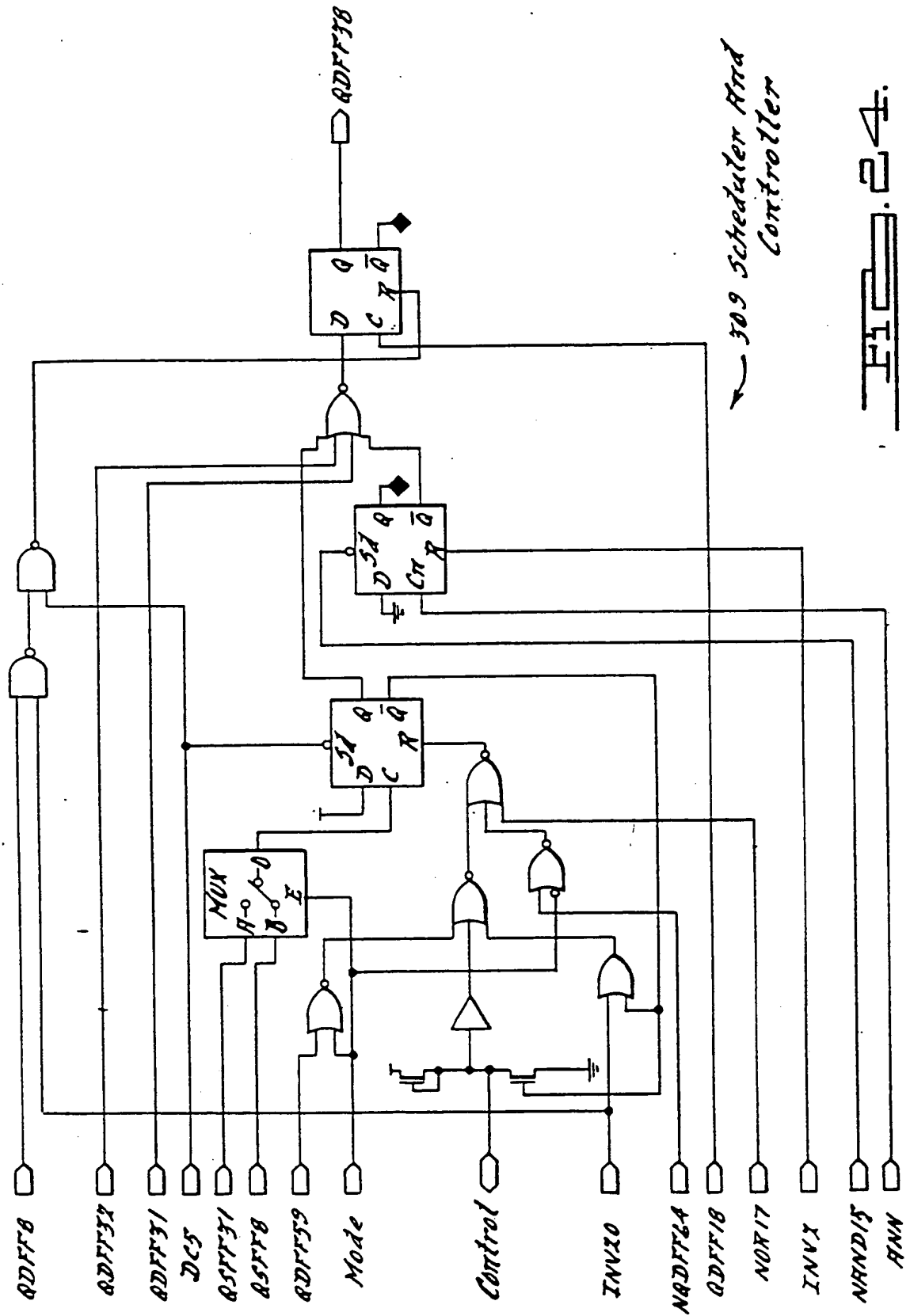
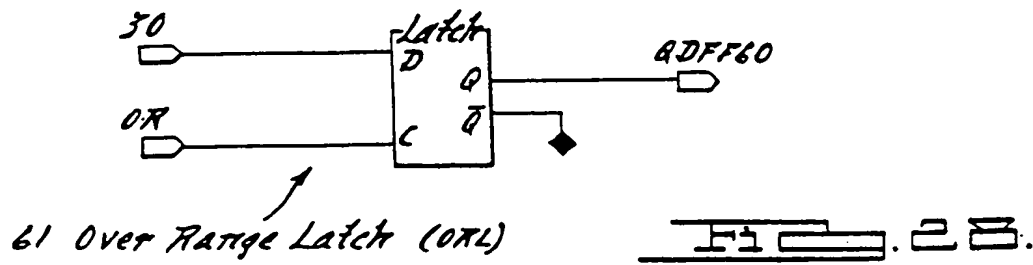
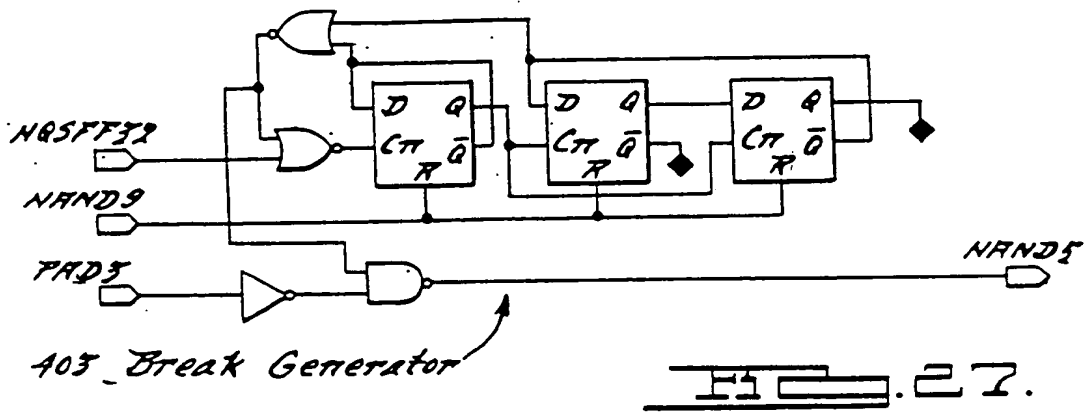
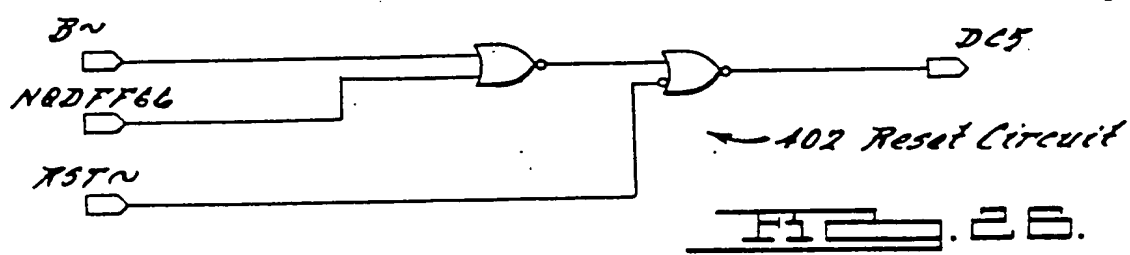
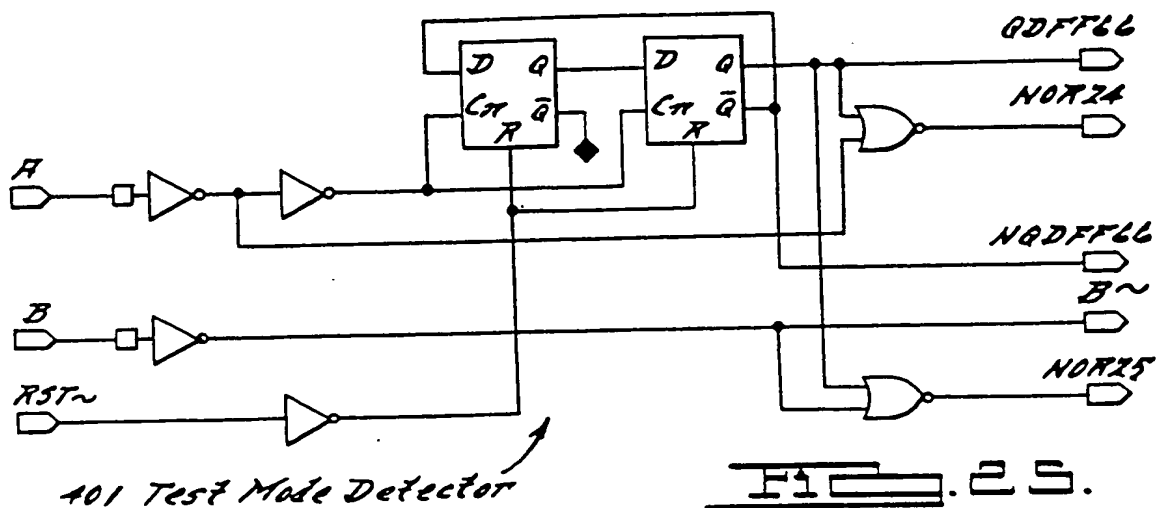
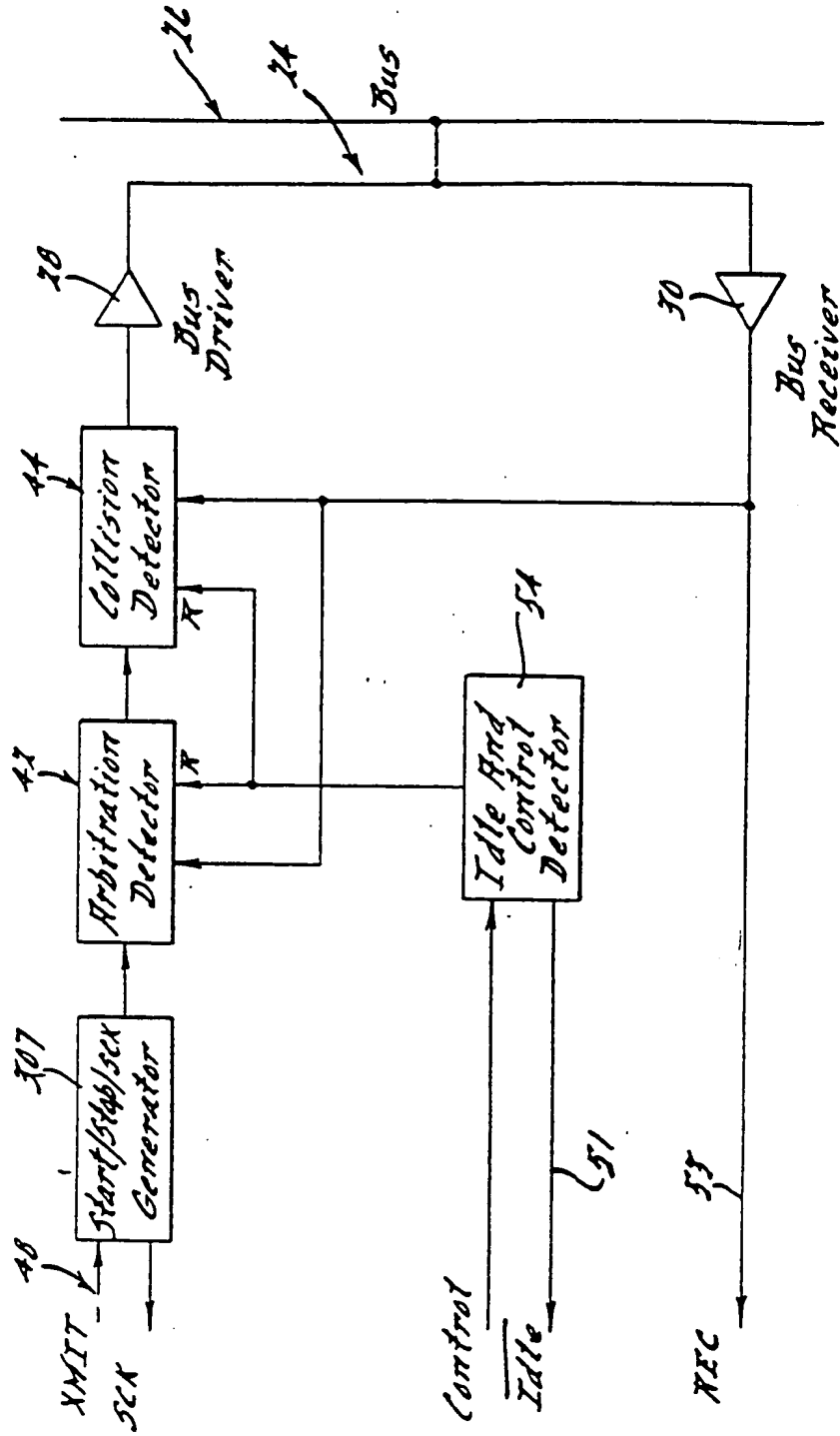


FIG. 24.

15/82



16/22



Unbuffered SPI

17/82

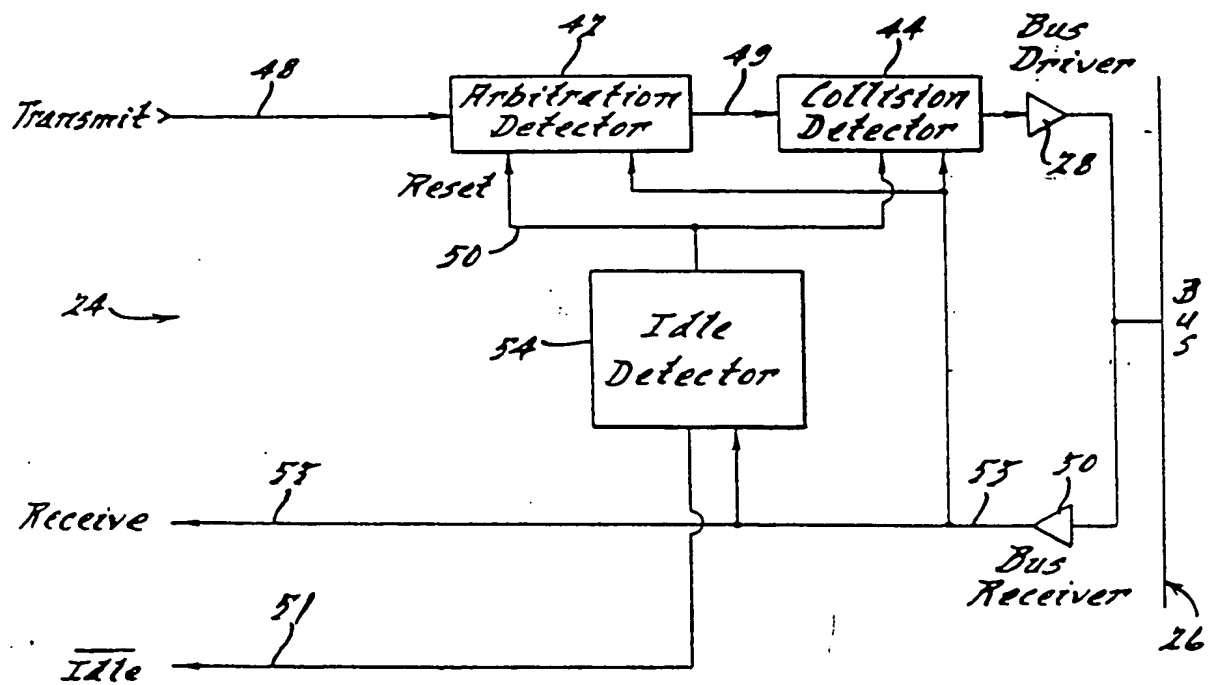
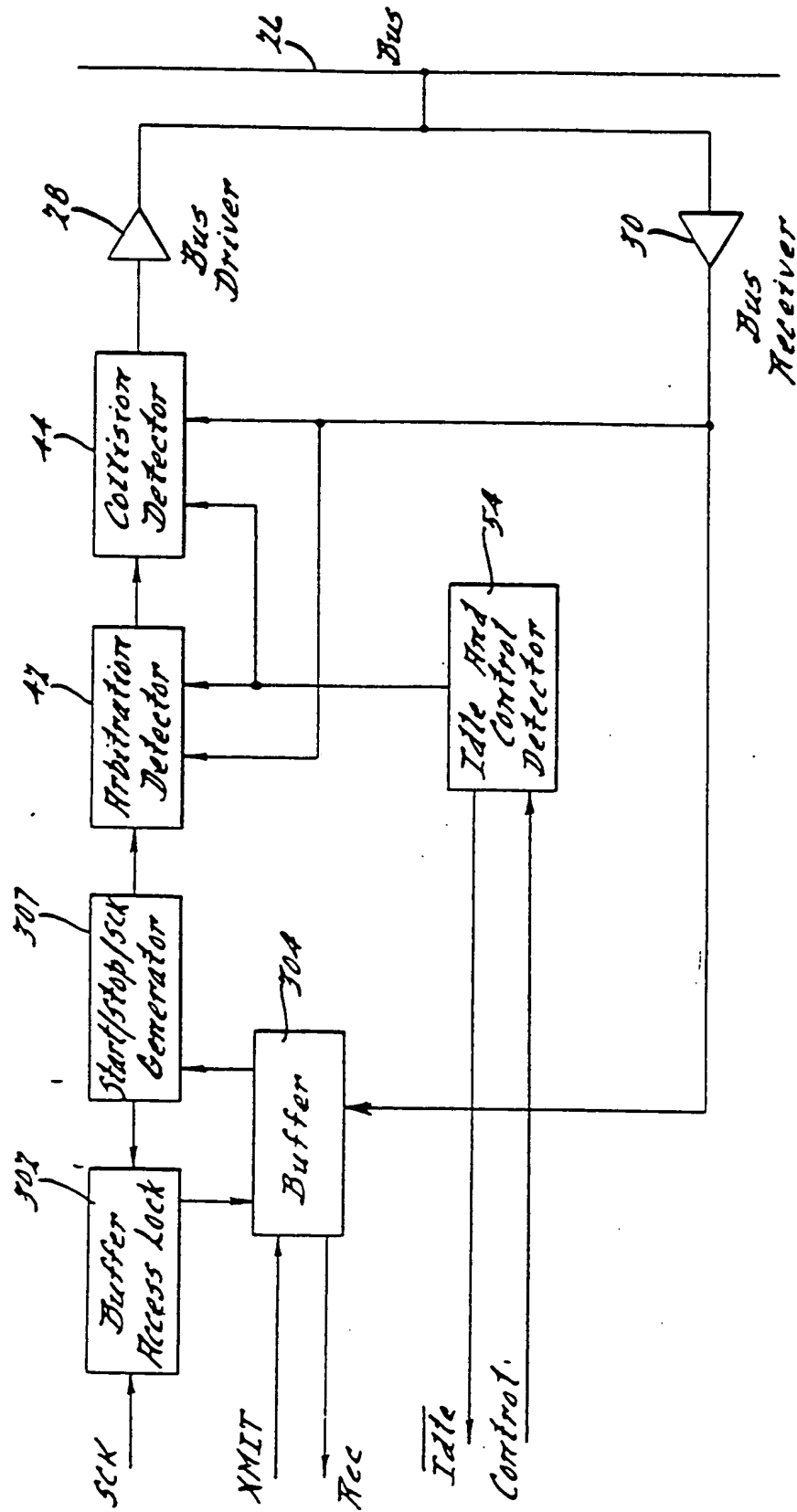


FIG. 30

18/22



Buffered SPI

FIG. 31

19/22

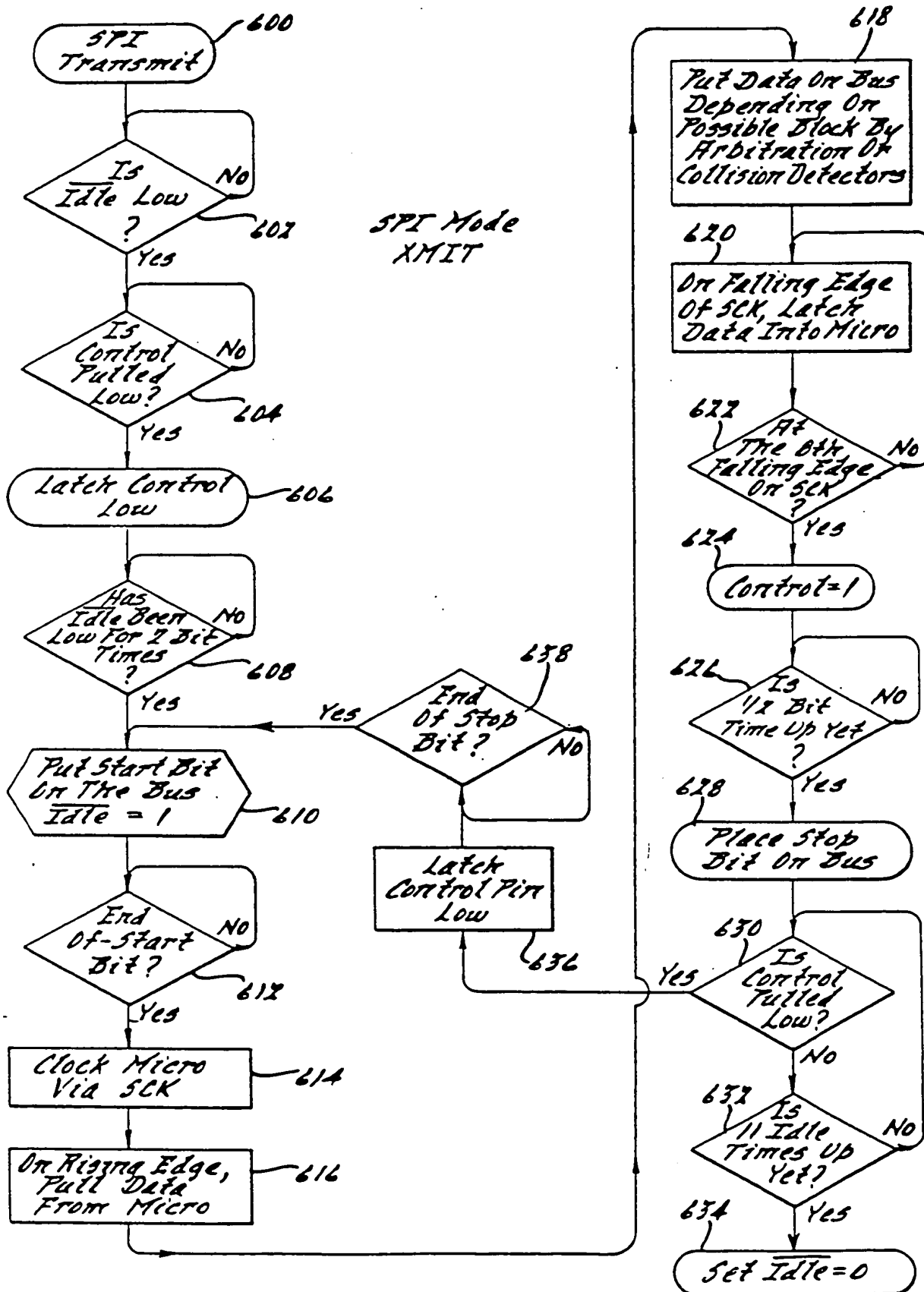
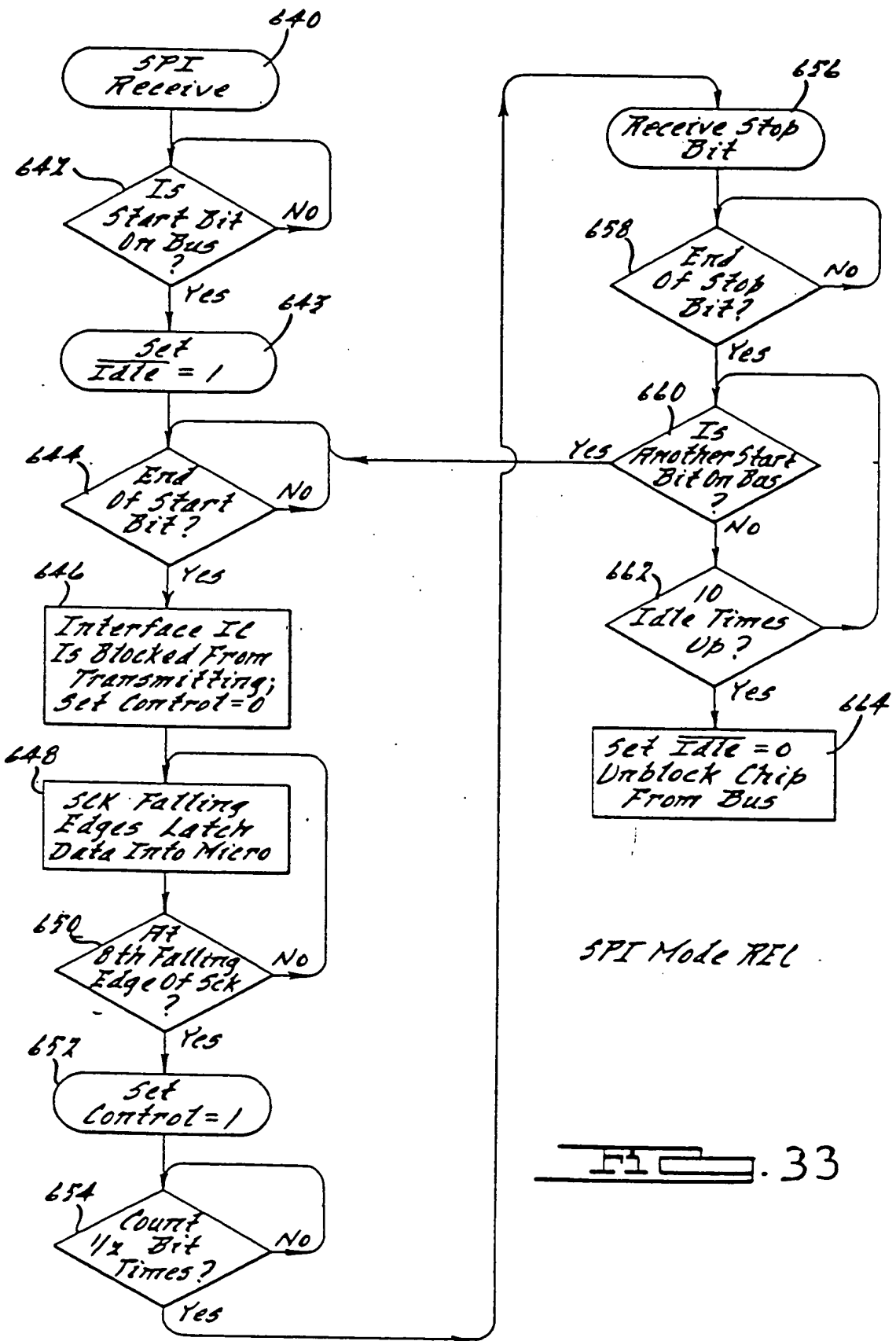


FIG. 3.2



SPI Mode RXC

FIG. 33

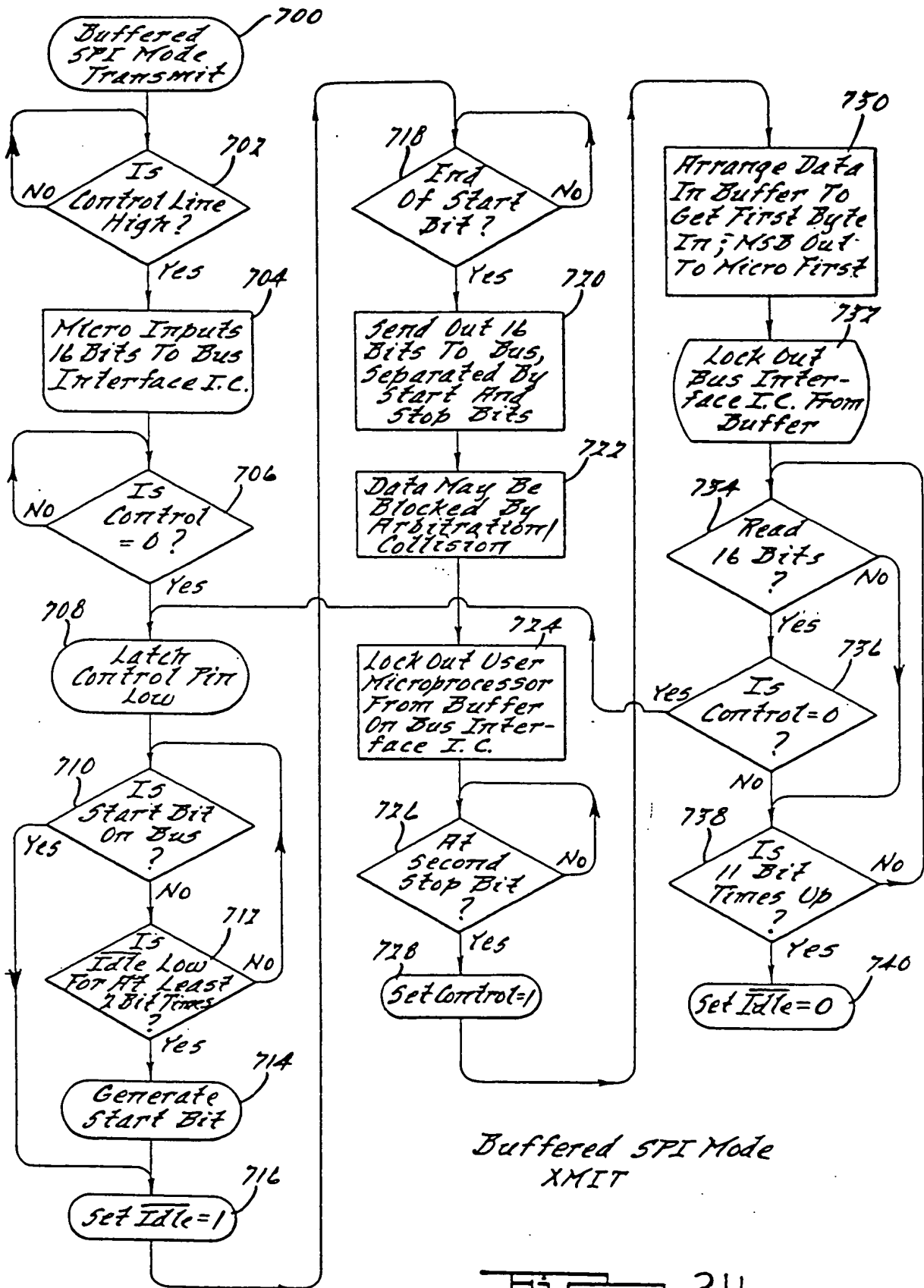


Fig. 34

